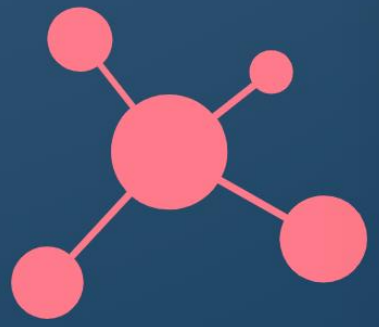
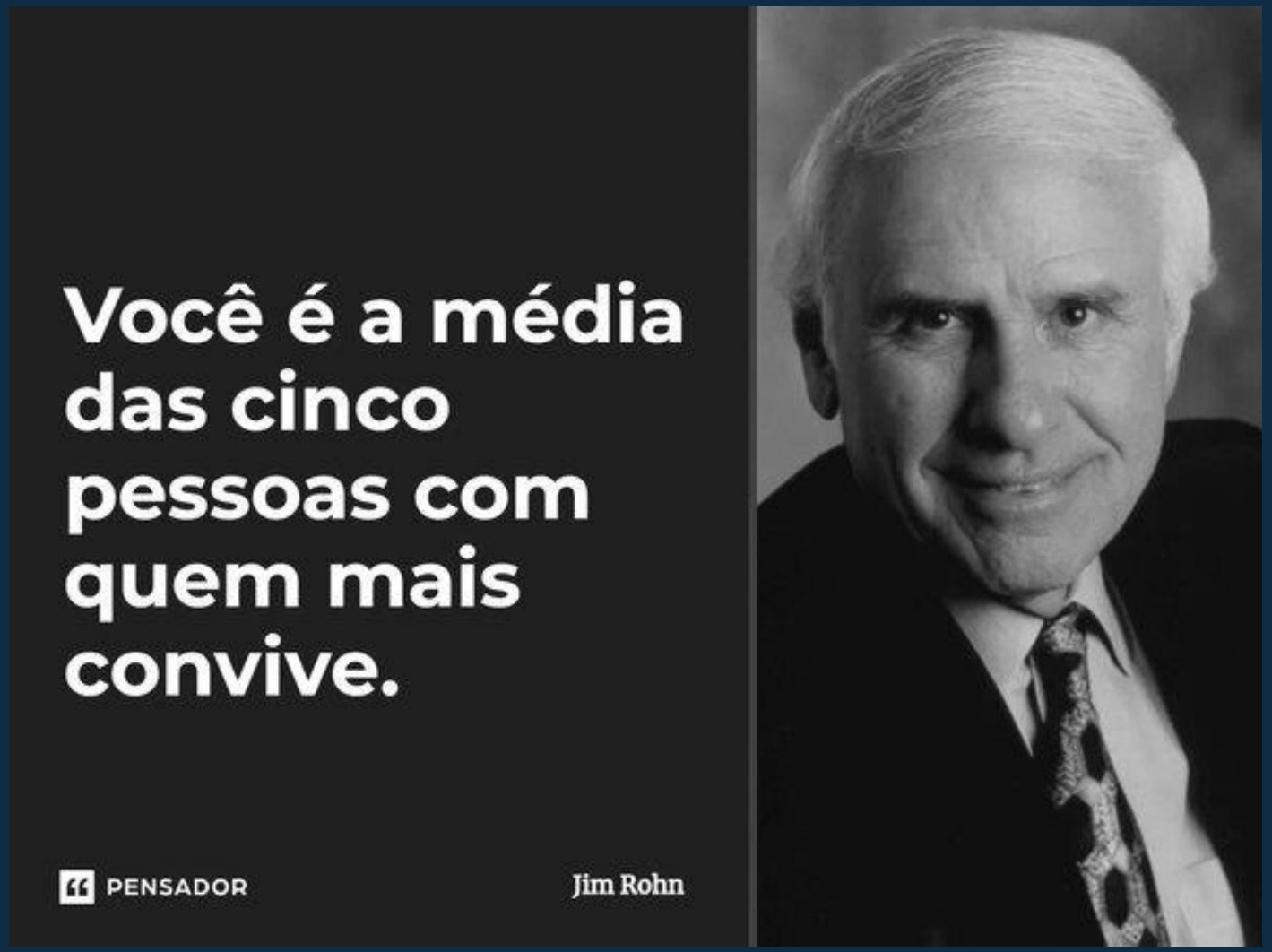
Tech&Dados
BY @ITALLEIRAS



Classificação - KNN
K-Nearest Neighbors

Funcionamento, exemplos, código e mais.

Agosto de 2023



E o que isso tem a ver com KNN?

Imagine um base de dados histórica em que temos duas variáveis de pessoas:

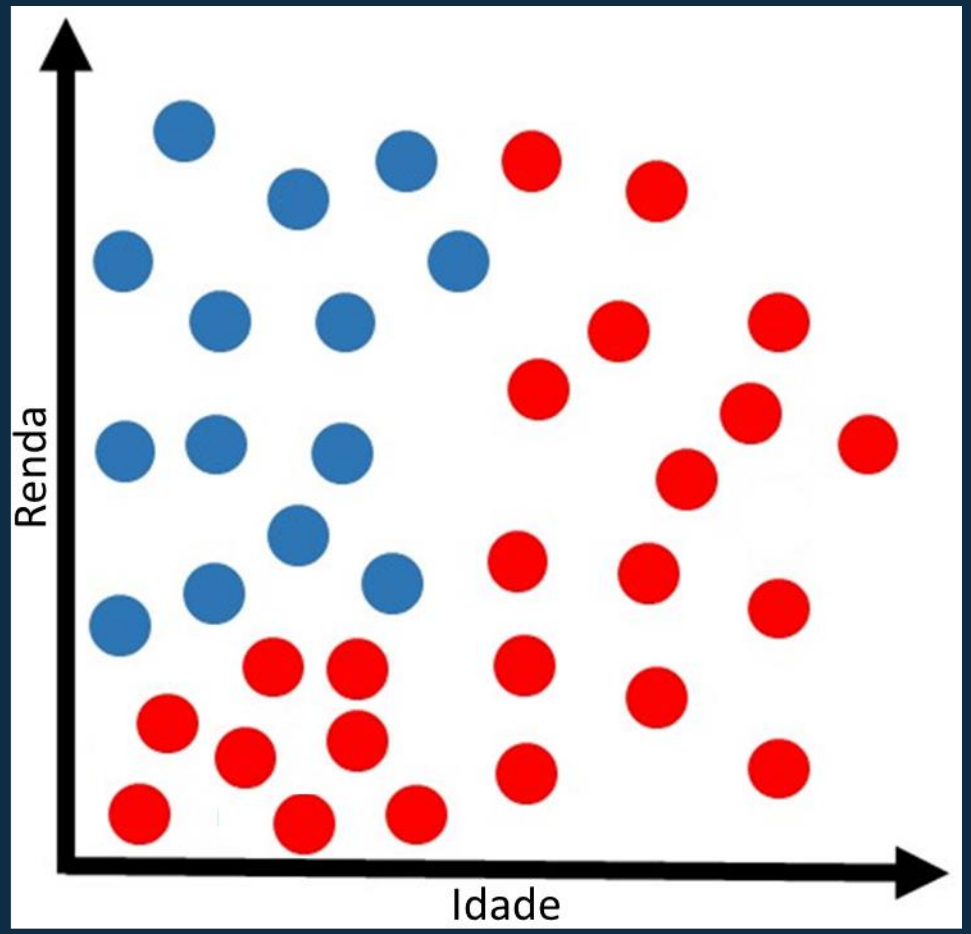
- Renda
- Idade

E temos uma variável target:

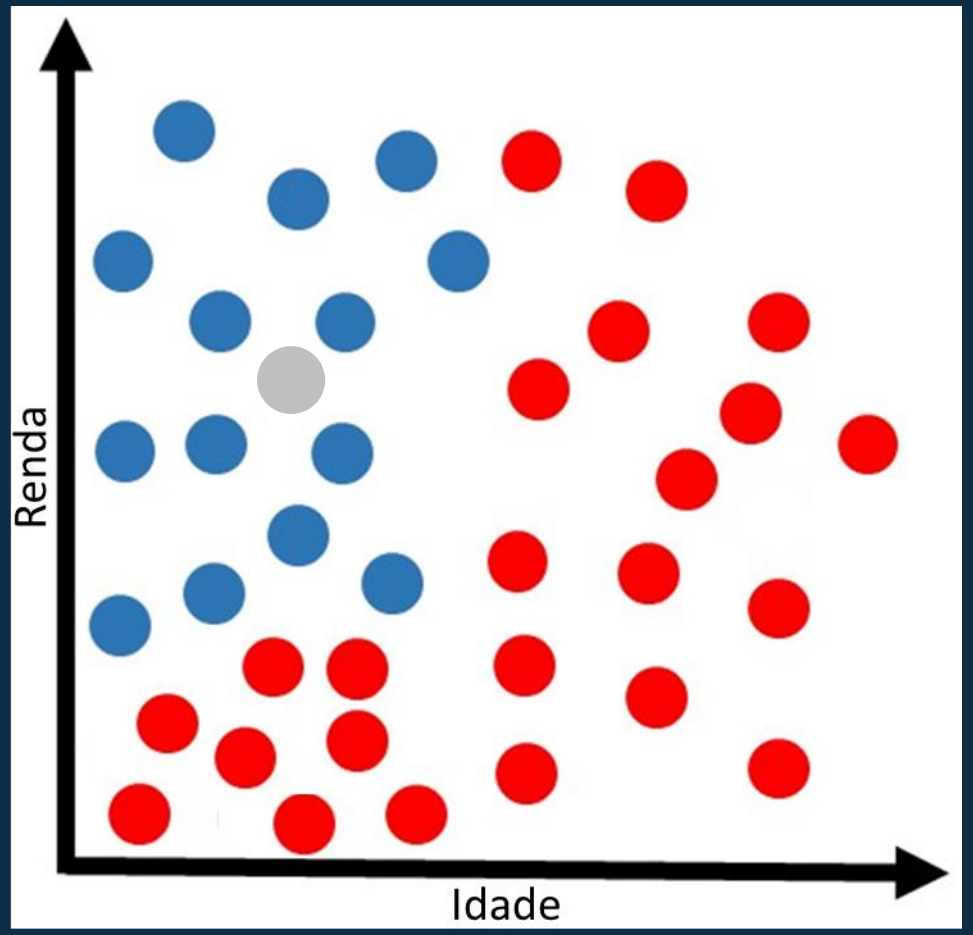
Comprou determinado produto

NÃO Comprou determinado produto

Nosso objetivo é construir um modelo para identificar potencias novos clientes.



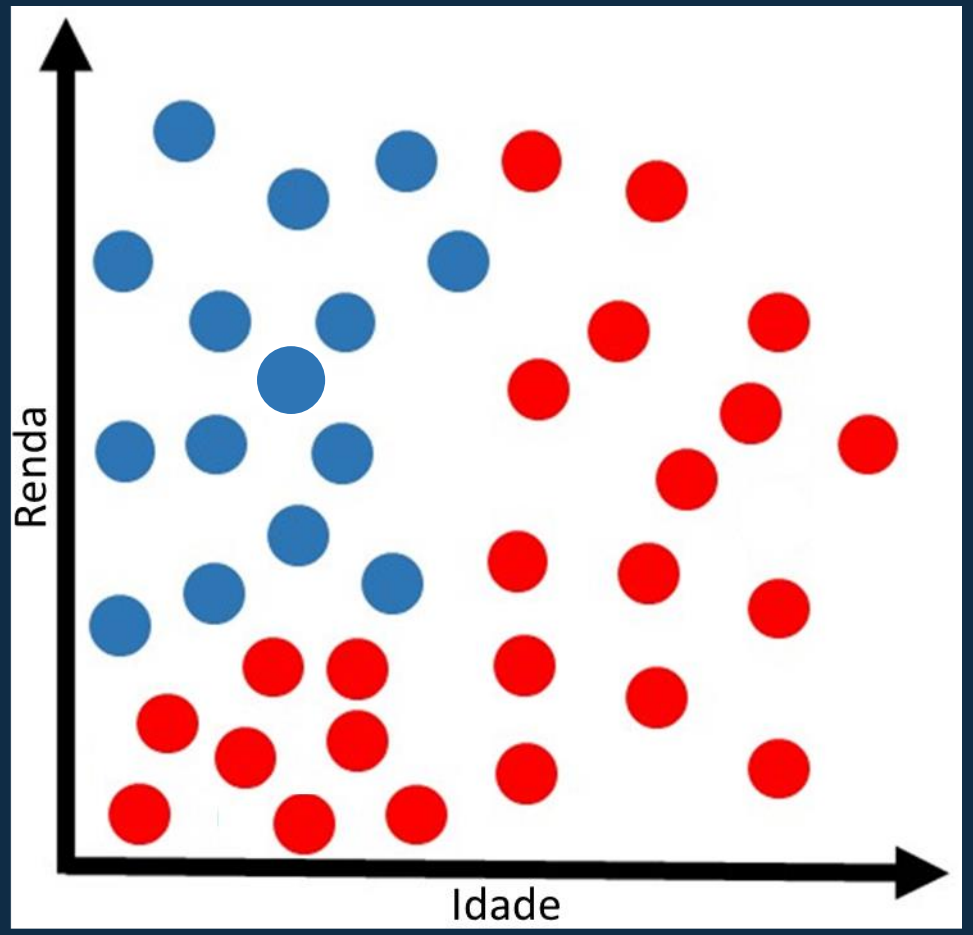
Como você acha que um novo objeto (uma nova pessoa), representada pela bolinha cinza, seria classificado pelo modelo?



Comprou determinado produto

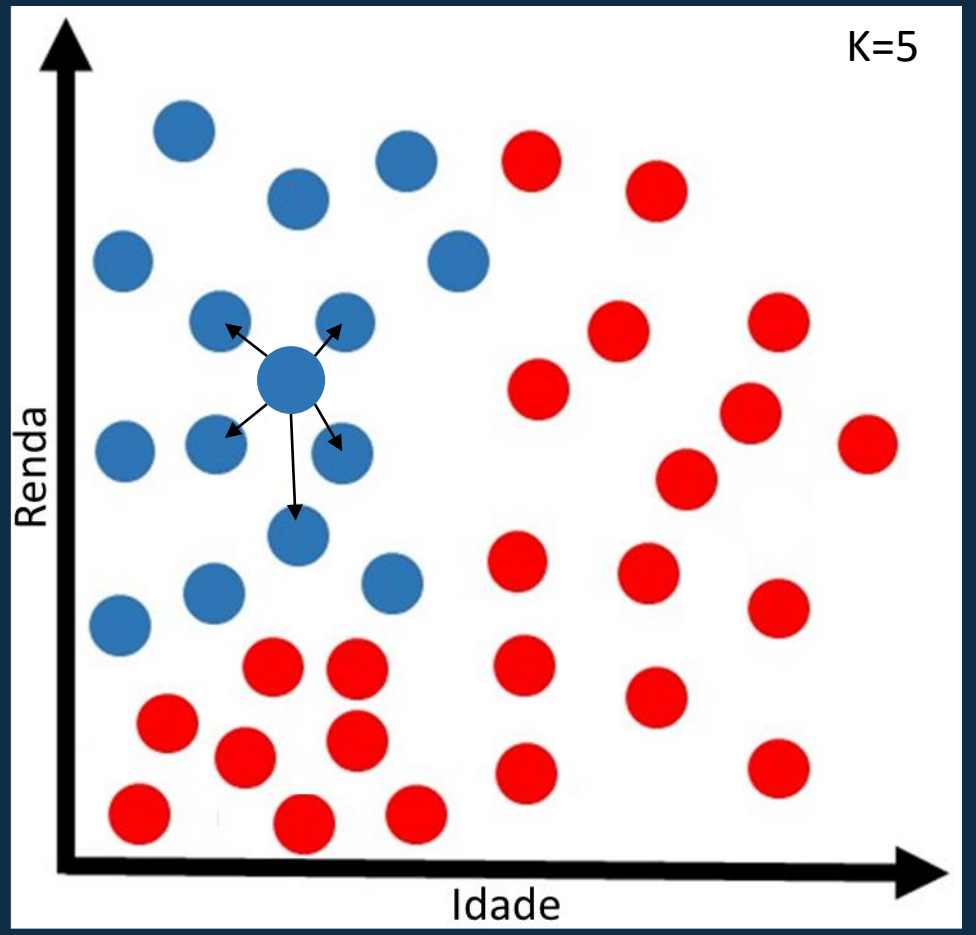
NÃO Comprou determinado produto

Intuitivamente, você deve dizer que o novo objeto seria classificado como “compra” o produto, na cor Azul.



É exatamente este o princípio do KNN. Objetos mais próximos definem a decisão do modelo.

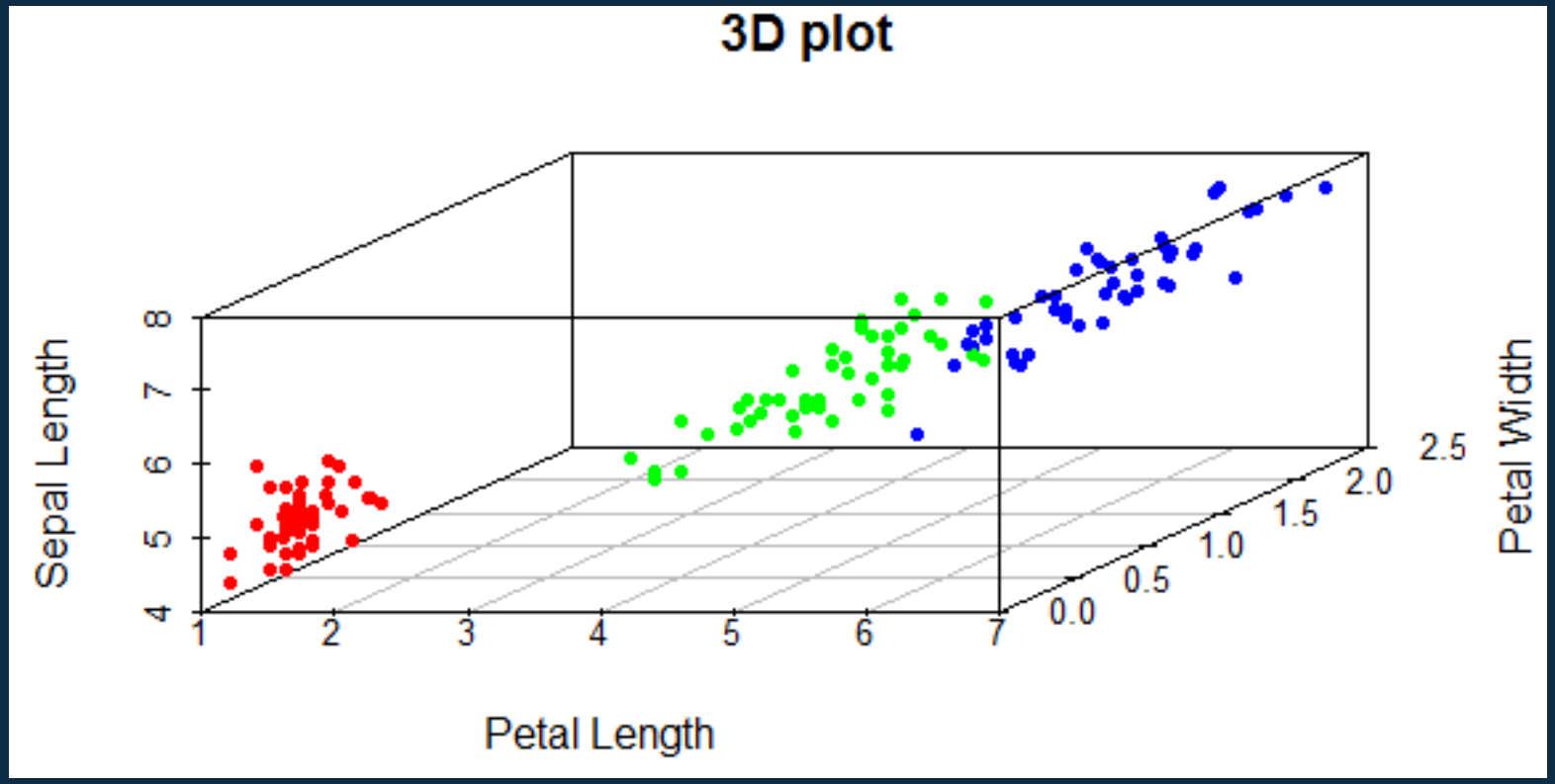
Por isso o nome K-Nearest Neighbors (K vizinhos mais próximos)



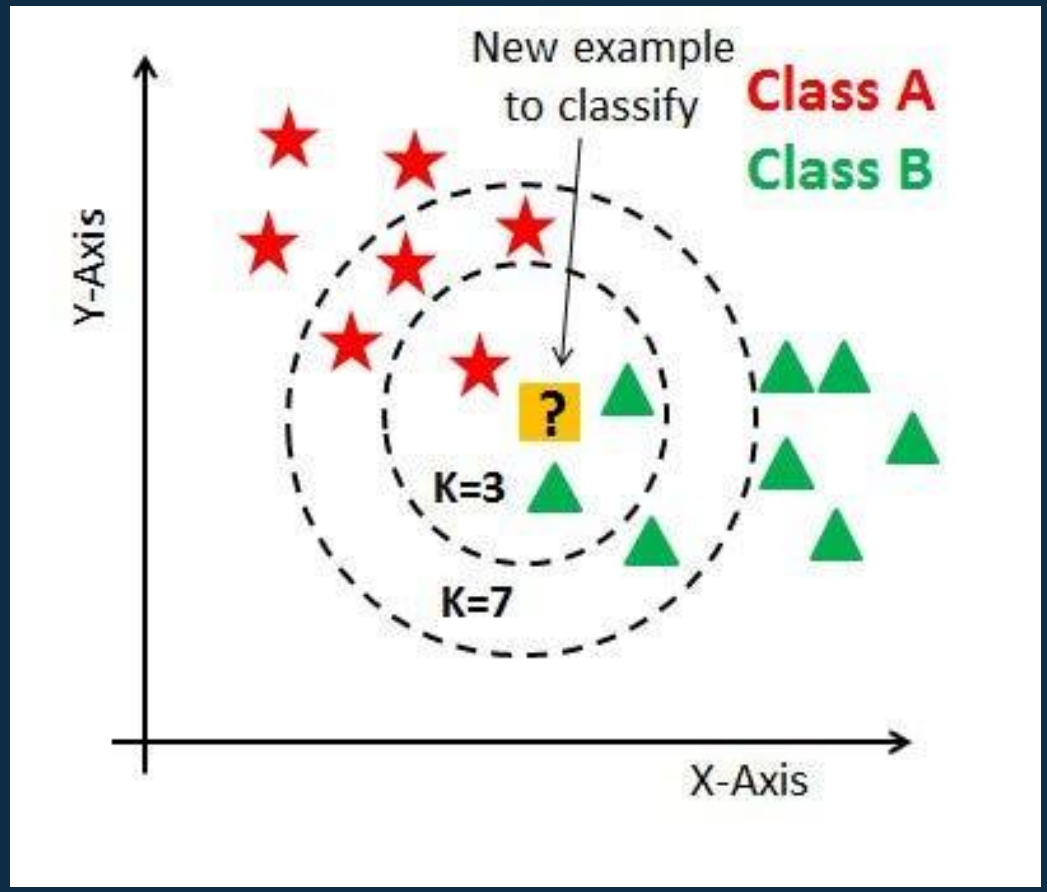
Comprou determinado produto

NÃO Comprou determinado produto

E não importa o número de variáveis no nosso problema. O KNN vai buscar pelos K vizinhos mais próximos para “escolher” a classe de um novo objeto.



Para isso, precisamos escolher o número de **K** (quantos vizinhos mais próximos serão considerados).



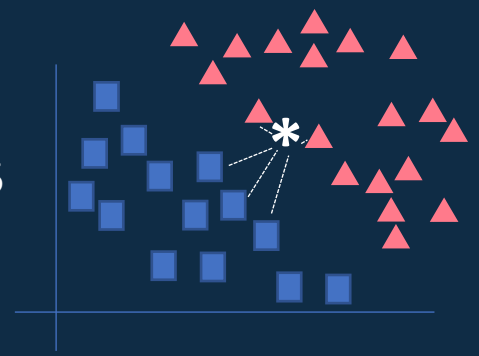
A votação pode ser feita de duas formas:

- Maioria de votos
- Peso ponderado pela distância

Dica: números ímpares de K evitam empates na votação 😉

Fonte imagem: https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb

Como ponderar o peso do objeto pela distância?



$\sum 1/d$ ■

$\sum 1/d$ ▲

O maior valor será escolhido como classe

■	Distância = 4	0,25 + 0,25 + 0,25 = **0,75**
▲	Distância = 2	0,5 + 0,5 = **1,00**

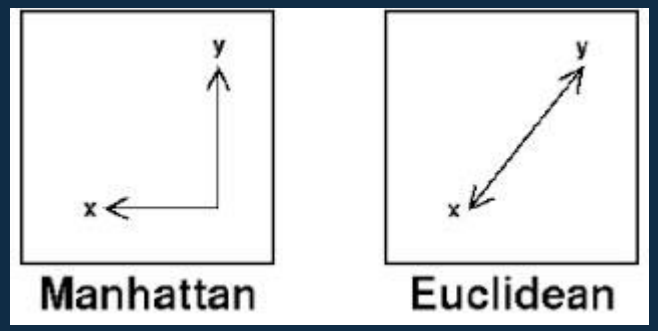
Classificação: ▲

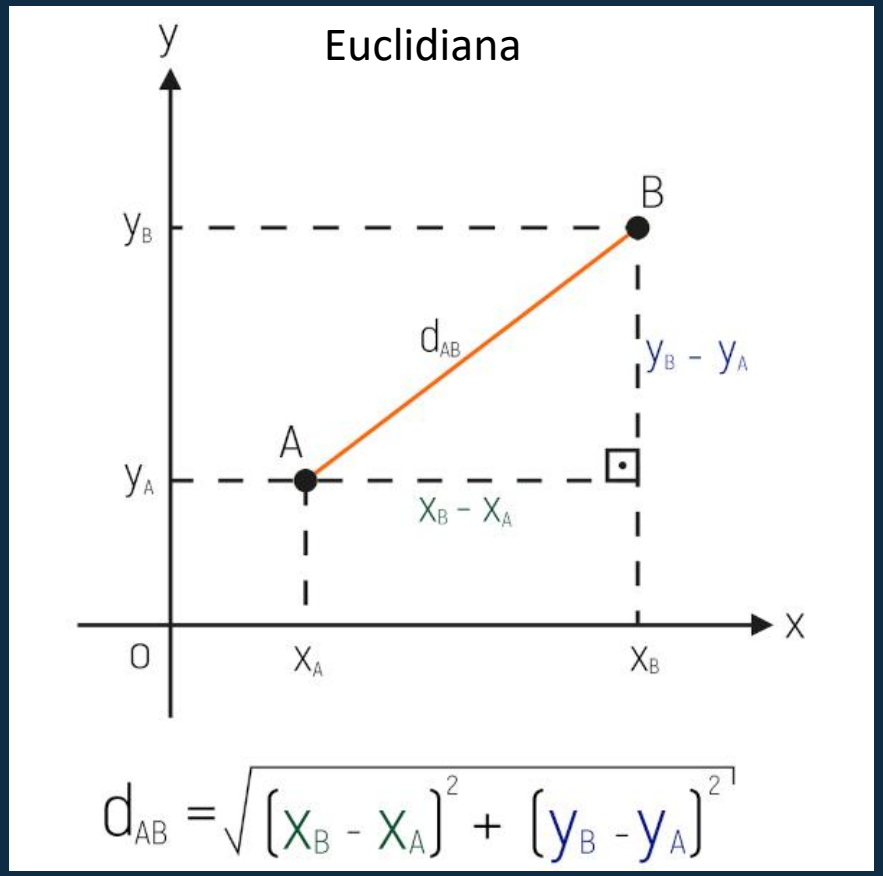
Também é possível atribuir um peso maior a objetos mais próximos. Para isso, deve-se utilizar o inverso da distância (1/d);

Como a distância é calculada?

Através de algumas métricas de distância.
As mais usadas são:

- Euclidiana
- Manhattan

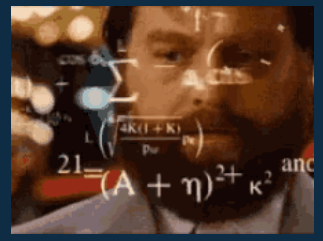




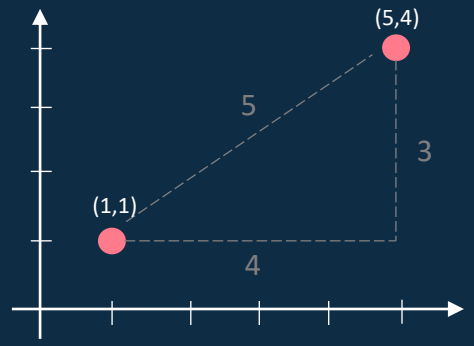
Como a distância é calculada?

Geralmente, a distancia também pode ser dada pela fórmula de Minkovisk.
Distância de Minkovisk: é uma generalização de distâncias. Com base no parâmetro p, é possível chegar em outras distâncias. Por exemplo, parâmetro p=1 torna a distância par Manhattan, p=2 torna a distância Euclidiana.

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$



Exemplo:

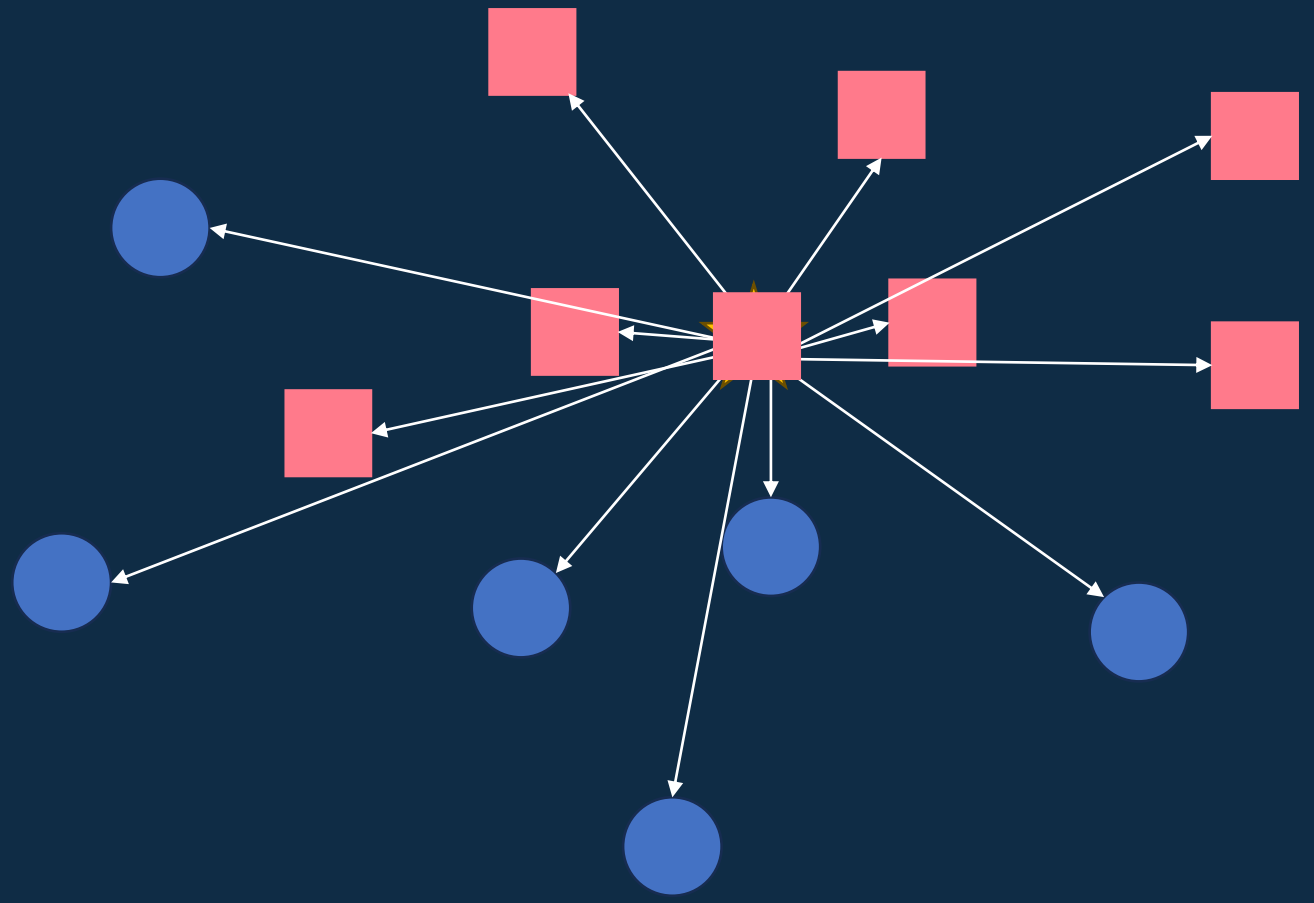


P=1 (Manhattan):

$$distancia = (|5-1|^1 + |4-1|^1)^{\frac{1}{1}}$$
$$distancia = |5-1| + |4-1|$$
$$distancia = 4 + 3$$
$$distancia = 7$$

P=2 (Euclidean):

$$distancia = (|5-1|^2 + |4-1|^2)^{\frac{1}{2}}$$
$$distancia = \sqrt{|5-1|^2 + |4-1|^2}$$
$$distancia = \sqrt{4^2 + 3^2}$$
$$distancia = \sqrt{25} = 5$$



K=5

Distância	Classe
1,2	■
1,3	■
1,3	●
1,8	■
2,0	■
2,1	●
2,4	■
2,6	■
2,8	●
3,0	■
3,2	●
3,2	●
3,4	●

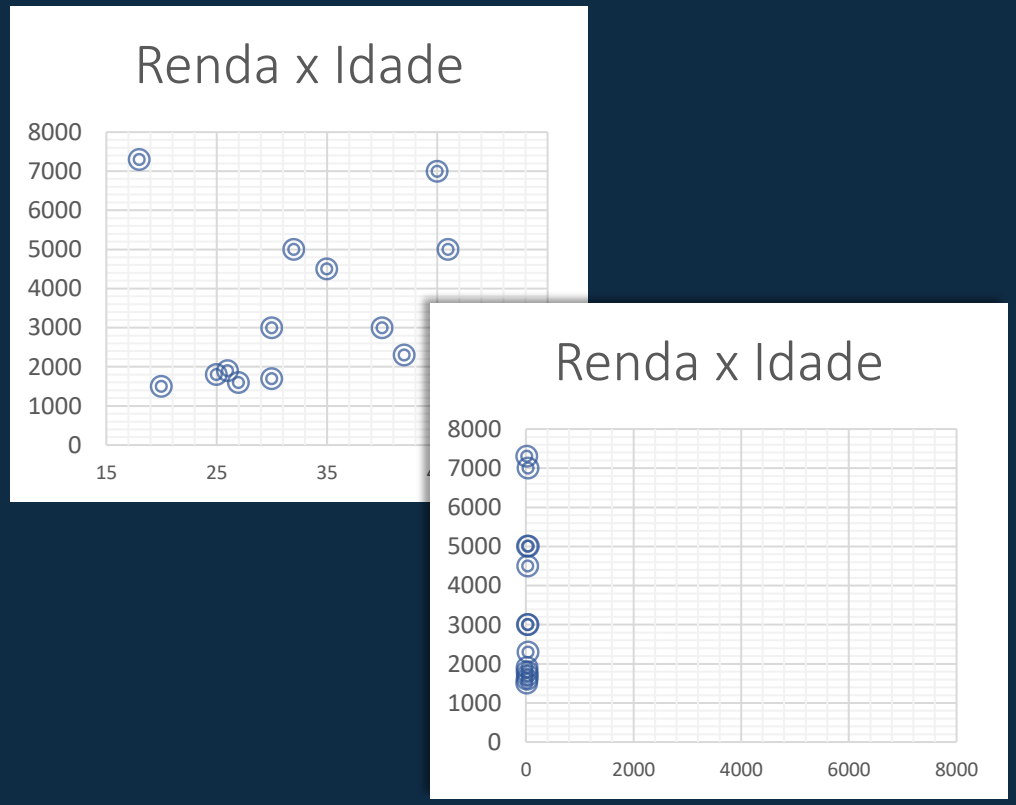
ATENÇÃO!!

O KNN é um algoritmo baseado em distâncias.

Portanto, as variáveis precisam ser:

- **Numéricas**
  - *One Hot Encoder , Ordinal Encoder*
- **Padronizadas / Normalizadas**
  - *StandardScaler, MinMaxScaler*

Normalizados? Veja porque:



Vantagens

- Não necessita treinar o modelo, apenas armazena em memória os dados de entrada (algoritmo lazy);
- É um algoritmo simples e aplicável mesmo em problemas complexos;
- É naturalmente incremental;
- Pode ser usado para problemas de regressão com poucas alterações (média dos k vizinhos);
- É um algoritmo não paramétrico (não necessita gerar uma função para chegar ao resultado).

Desvantagens

- Muitas dimensões tornam o algoritmo mais complexo;
- A escolha do K é difícil;
- Sofre com problemas de escala.
- A predição é uma etapa computacionalmente custosa;

Solução: Utilizar técnicas de particionamento do espaço multidimensional (kd-tree, ball-tree).

Próximo vídeo!!

Modelo	KNN

Objetivo Resumido	O algoritmo tem por objetivo classificar um novo registro de entrada com base nos k vizinhos mais próximos (mais semelhantes). Cada objeto de entrada representa um ponto em um espaço n-dimensional. Os objetos rotulados são armazenados em memória e, sempre que é necessário uma classificação de um novo registro, o algoritmo calcula a distância deste novo objeto para todos os demais objetos armazenados. Os **k** vizinhos mais próximos definirão, por voto majoritário, qual a classe do novo registro.

Tipo	Classificação
Categoria	Supervisionado

Premissas para o funcionamento	- Dentro do Scikit Learn, os dados precisam ser numéricos e normalizados (algoritmo baseado em distância).	Parâmetros	**K**: número de vizinhos a ser considerado na votação da classe **Peso:** Associa um peso à contribuição de cada vizinho (uniform, distance) **Algoritmo:** Forma de indexação dos dados durante o treino (KD-Tree, Ball-Tree e brute-force) **Metrica de distancia**: Define como será calculada a distância entre os vizinhos. Valor default = Minkowski. Alterar P para modificar (2 = Euclidean, 1 = Manhattan).

Funcionamento detalhado (Steps, informações detalhadas etc.)	O algoritmo inicia sendo definida a variável k (o número de vizinhos a ser considerado na votação). Os dados de treinamento (rotulados) são armazenados em memória e não gera um modelo. Quando um novo registro precisa ser classificado (predict), o algoritmo calcula a distância (geralmente euclidiana) do novo objeto para todos os demais objetos armazenados e escolhe os k objetos mais próximos. O parâmetro K geralmente é ímpar para não ocorrer empates. O objeto será classificado com o rótulo da maior classe dentre os vizinhos. É possível também utilizar um peso à contribuição dos vizinhos, que beneficia os pontos mais próximos do objeto a ser classificado.	**STEPS:** 1. Definir o parâmetro K (instanciar o modelo); 2. Armazenar os dados rotulados em memória (método Fit - algoritmo lazy); 3. Ao receber um novo objeto para classificar, escolhe aqueles K objetos mais próximos ao novo objeto com base na distância. 4. A classe do objetos que mais se repetirem dentre os K vizinhos será a classe do novo objeto. 5. Caso seja empregado peso na votação baseado na distância, é realizada a soma do inverso da distância para cada classe. A maior soma classificará aquele objeto.

Vantagens	- Não necessita treinar o modelo, apenas armazena em memória os dados de entrada (algoritmo lazy) - É um algoritmo simples e aplicável mesmo em problemas complexos. - É naturalmente incremental - Pode ser usado para problemas de regressão com poucas alterações (média dos k vizinhos) - É um algoritmo não paramétrico (não necessita gerar uma função para chegar ao resultado)	Desvantagens	- A predição é uma etapa computacionalmente custosa. - Muitas dimensões tornam o algoritmo mais complexo. - A escolha do K é difícil - Sofre com problemas de escala

Como avaliar o desempenho	- O desempenho durante o teste pode ser medido com cálculos de acurácia, precisão, recall, curva ROC etc.	Como corrigir ou compensar desvantagens	- Selecionar subconjuntos de atributos mais relevantes. - Reduzir a dimensionalidade dos atributos. - Utilizar técnicas de particionamento do espaço multidimensional (kd-tree, ball-tree).

Obrigado!

youtube.com/@Tech_dados

linkedin.com/in/itallo-dias/