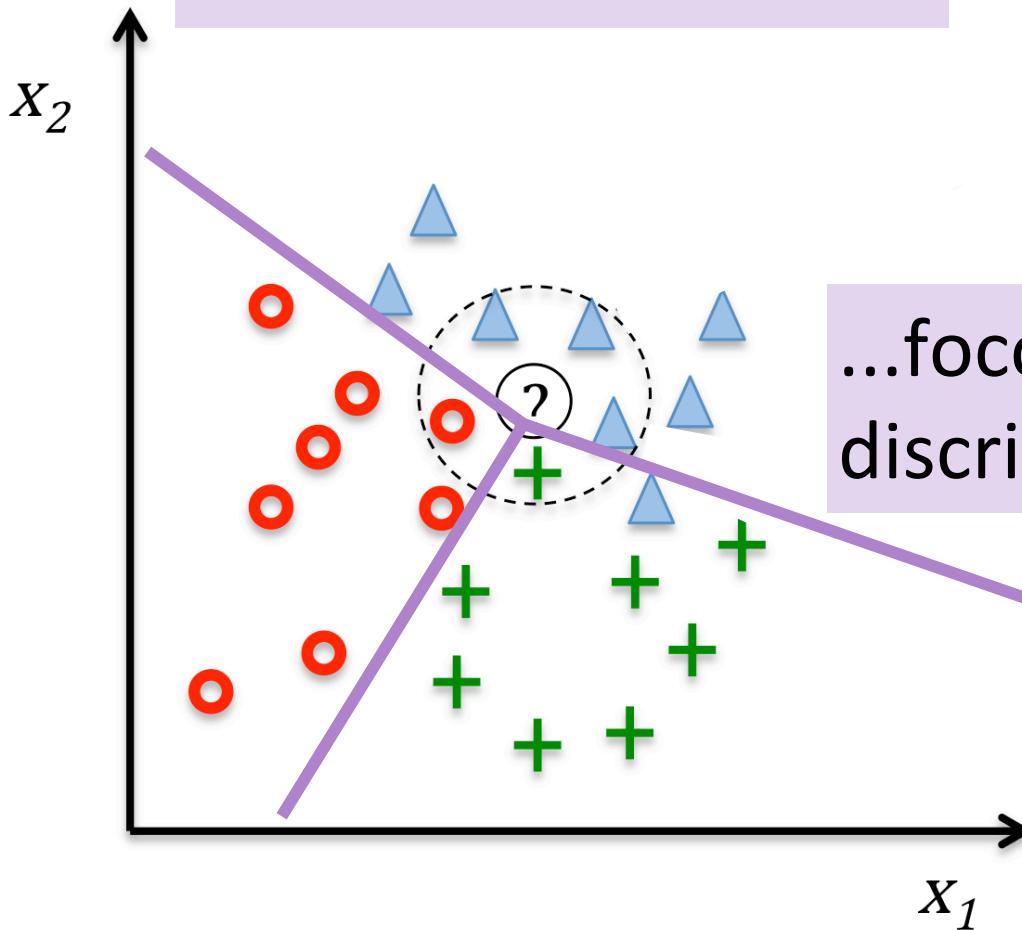
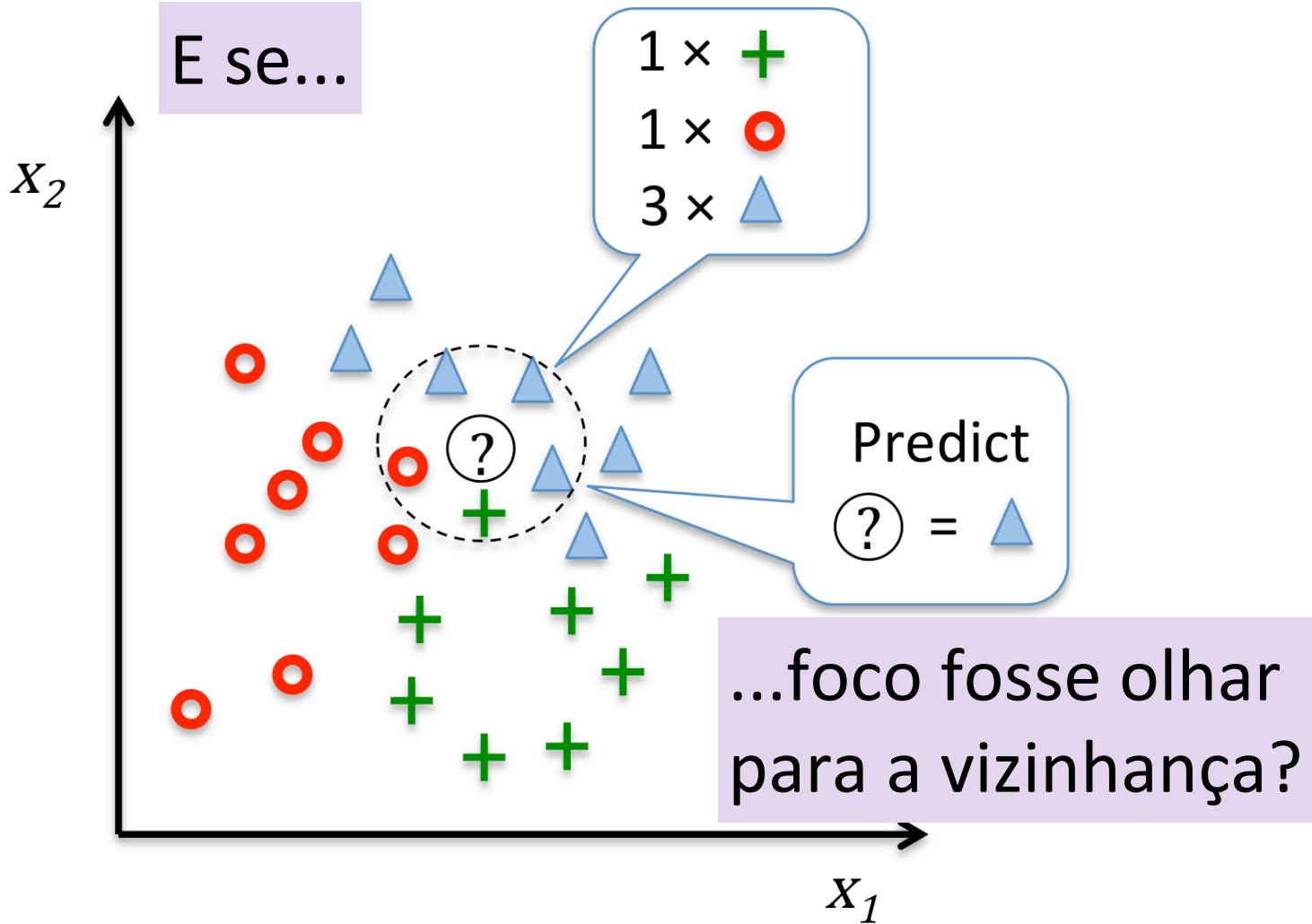


Modelos anteriores...



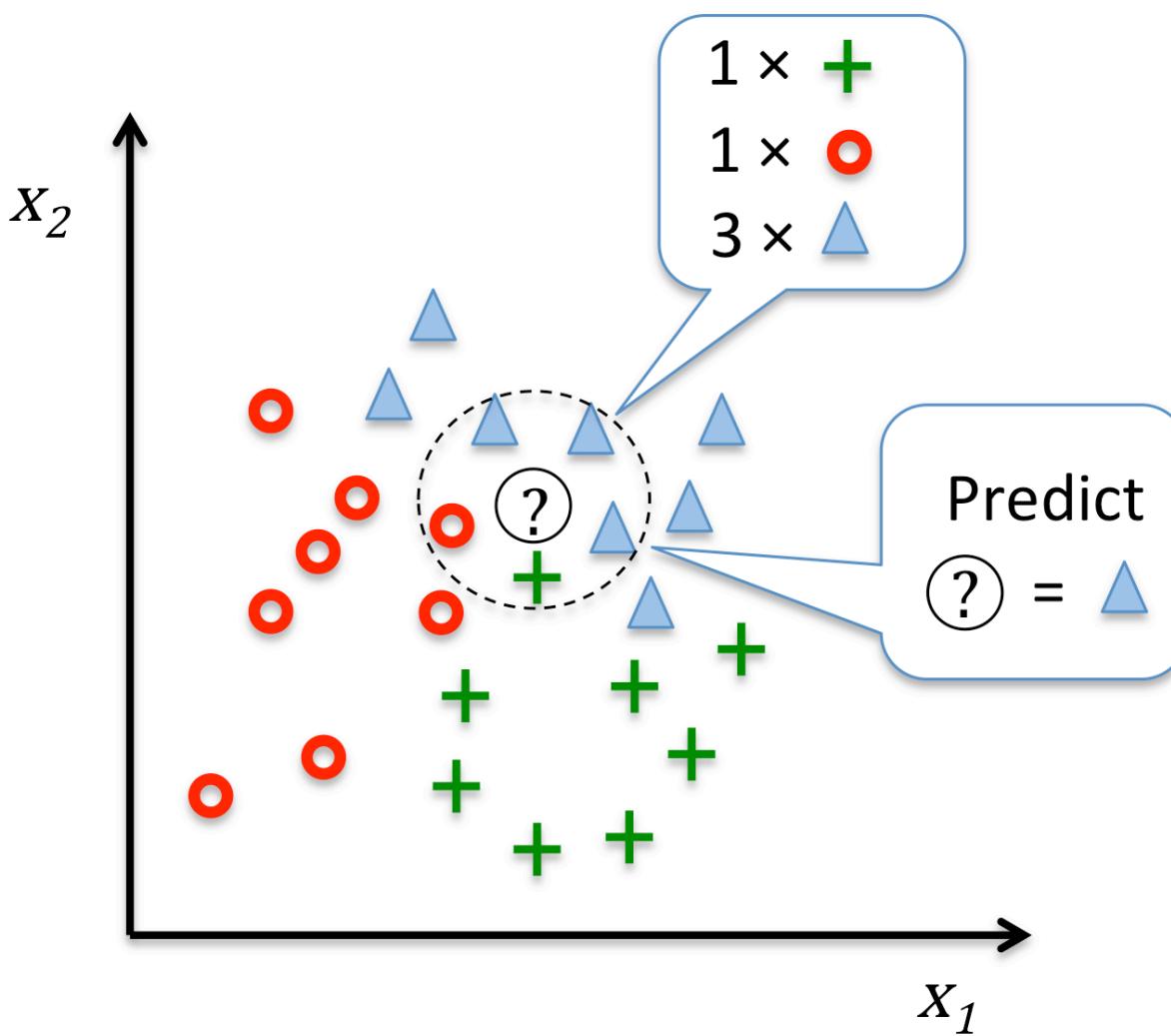
...foco na busca por
discriminantes!



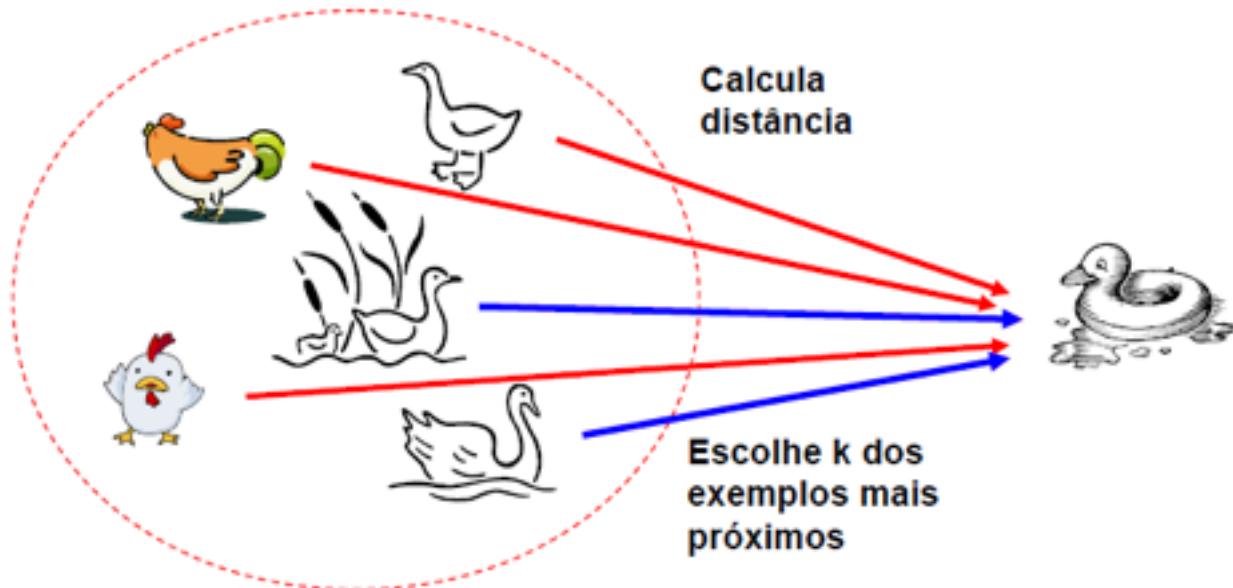
K Vizinhos mais Próximos (k-NN)

Métodos baseados em distância

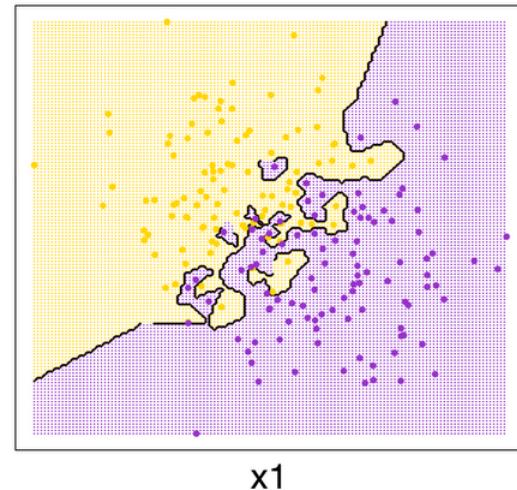
5-NN



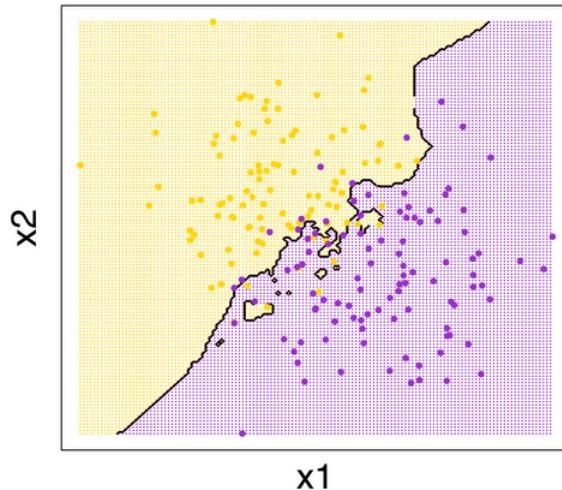
Princípio: se anda como um pato, e grasna como um pato, então provavelmente é um pato.



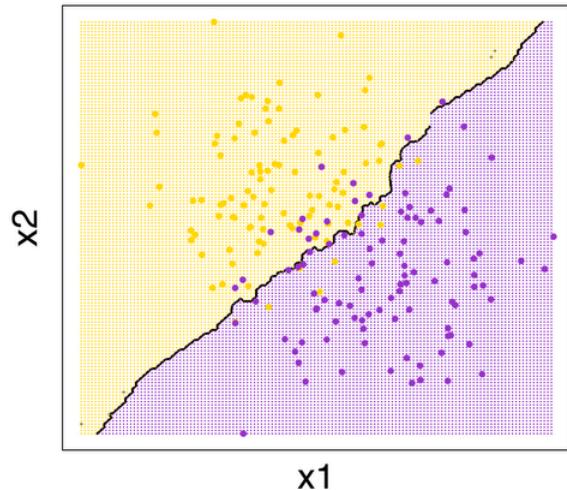
Binary kNN Classification (k=1)



Binary kNN Classification (k=5)



Binary kNN Classification (k=25)



nearest neighbour ($k = 1$)

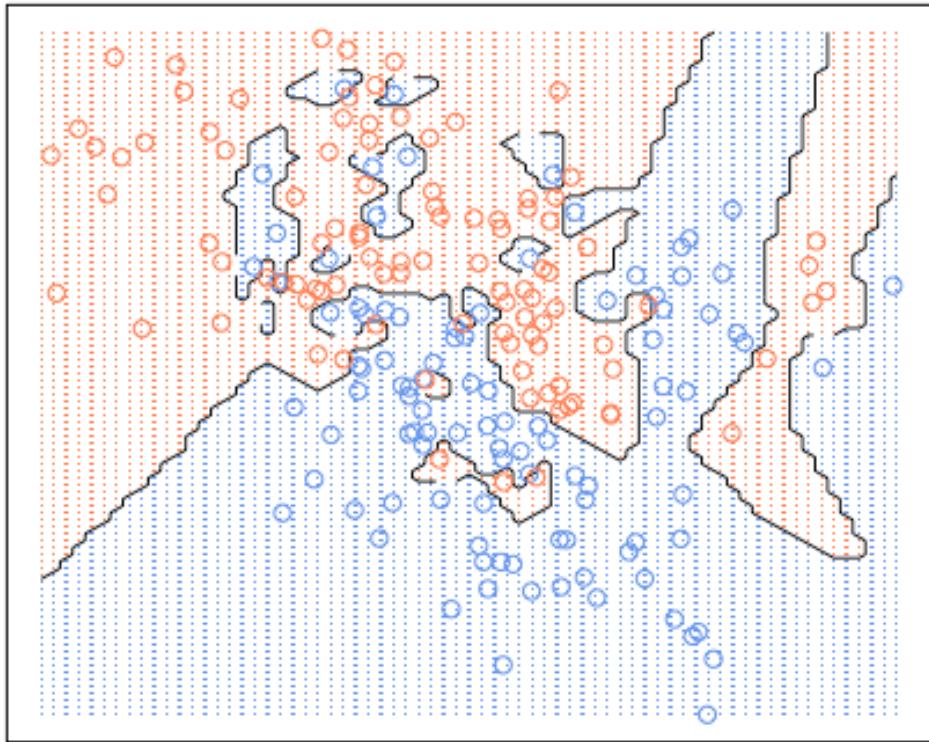
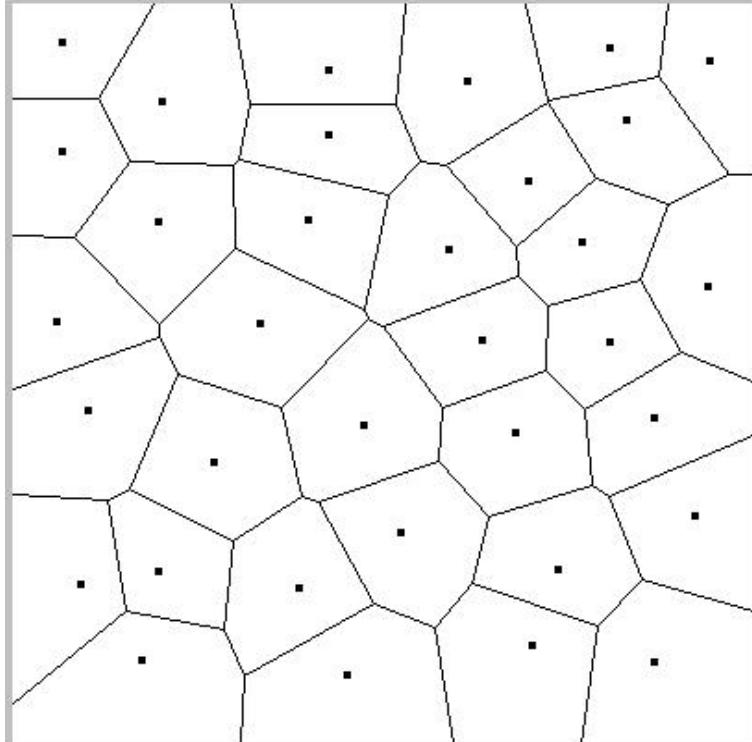
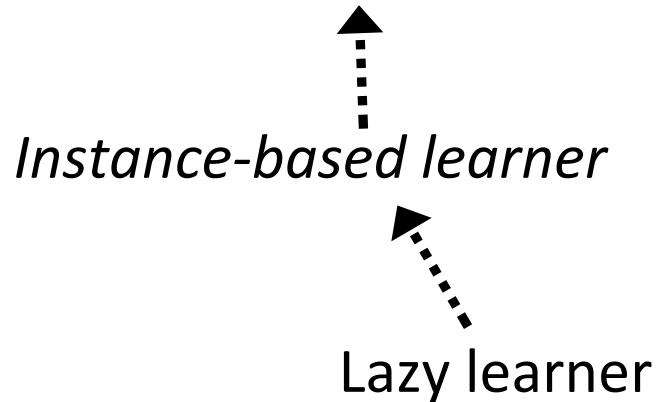


Diagrama de Voronoi



kNN

- Não aprende uma função discriminativa a partir dos dados de treinamento
- Deixa o trabalho de classificação para o momento em que for exigido (uso)
- *Modelo não paramétrico* vs. *Modelo paramétrico*



kNN

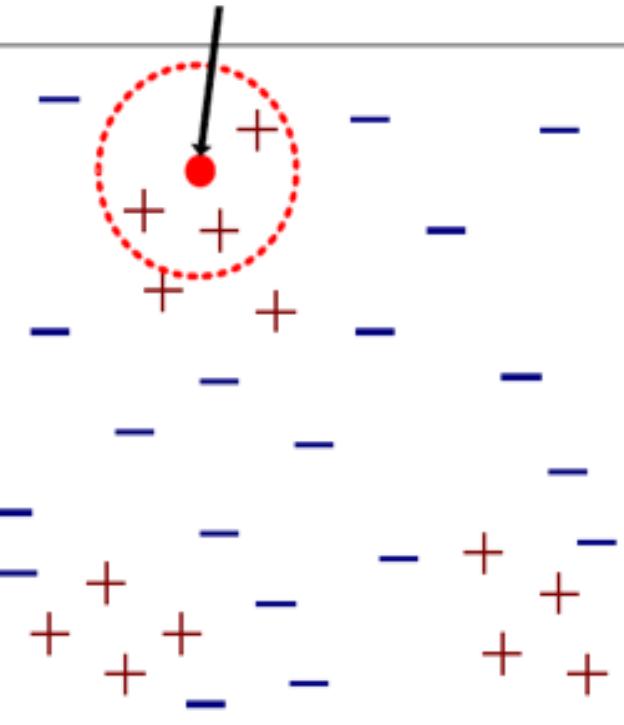
Algoritmo

1. Escolhe um valor de k e **medida de distância**;
2. Encontra os k vizinhos mais próximos do novo ponto (exemplo que se quer classificar);
3. Assinala o rótulo de classe por voto majoritário.

Medida de distância

Novo exemplo

$$Y = (y_1, y_2, \dots, y_d)$$



Métrica de Minkowski:

$$d(x, y) = \left(|x_1 - y_1|^p + |x_2 - y_2|^p + \dots + |x_d - y_d|^p \right)^{1/p}$$

p=1 → ?
p=2 → ?

...

Importante!

Reescalar os dados de modo que as features contribuam igualitariamente para a distância

Vantagens vs. Desvantagens

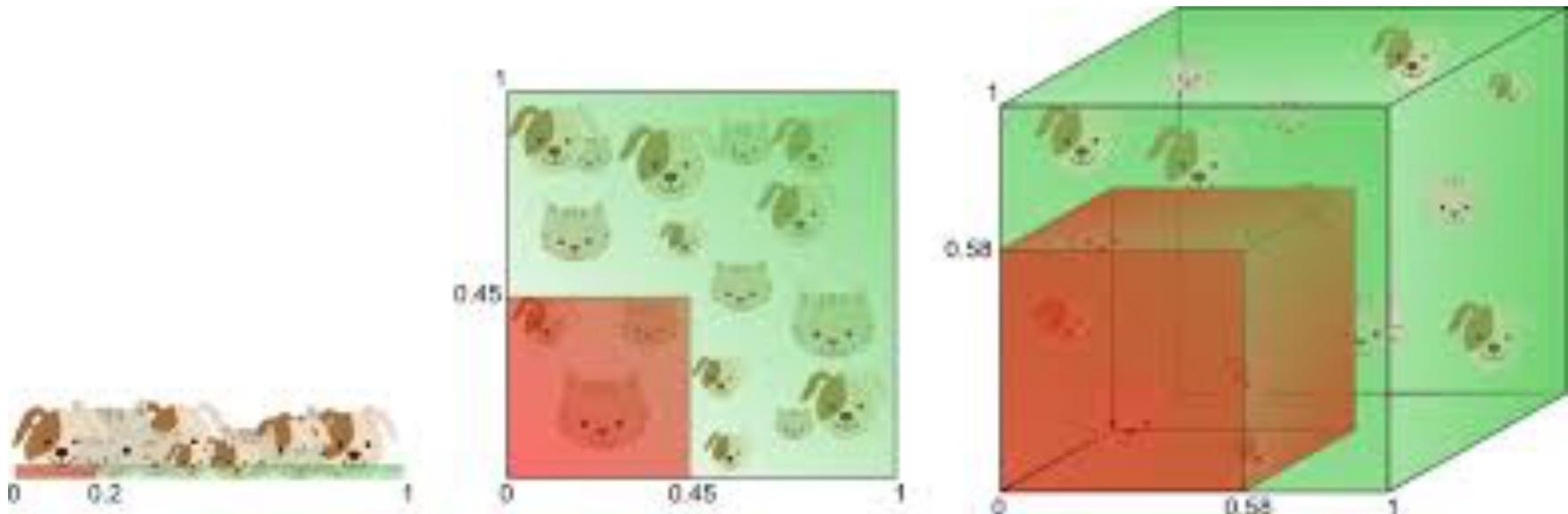
- Classificador adapta dinamicamente a novos dados de treinamento
- Complexidade computacional $O(d n)$, onde $d = \#features$, $n = \#exemplos$
- Espaço de armazenamento: usualmente não há descarte de dados!

Friedman, J. H., Bentley, J. L., & Finkel, R. A.
(1977). An algorithm for finding best matches in
logarithmic expected time. *ACM Transactions on
Mathematical Software (TOMS)*, 3(3), 209-226.



KD-Trees
 $O(d \log n)$

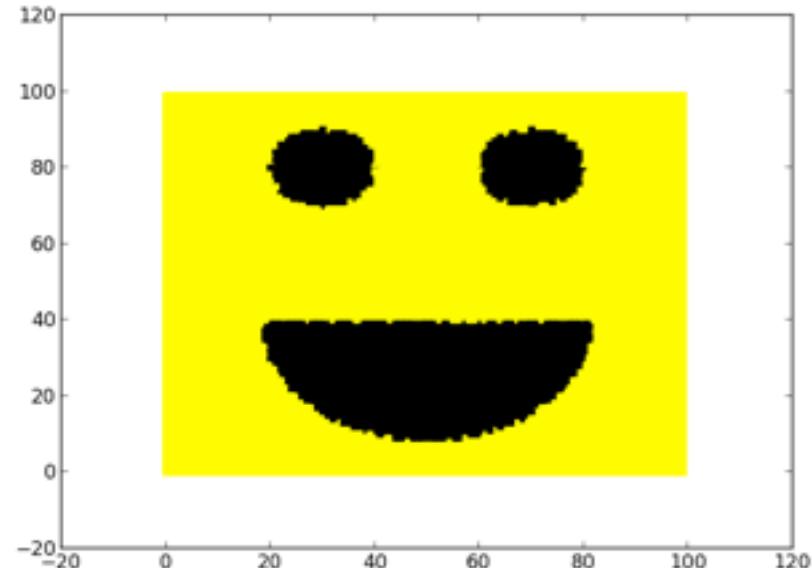
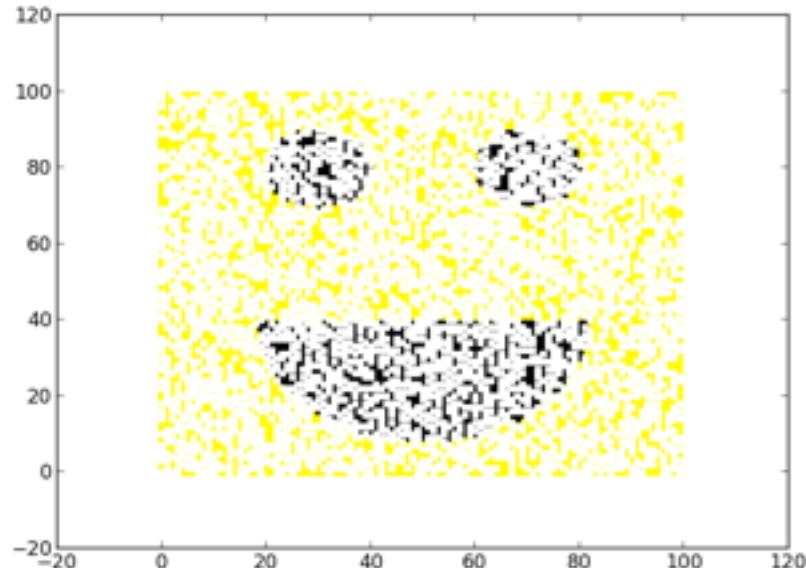
- Maldição da dimensionalidade



- esparsidade dos dados → +

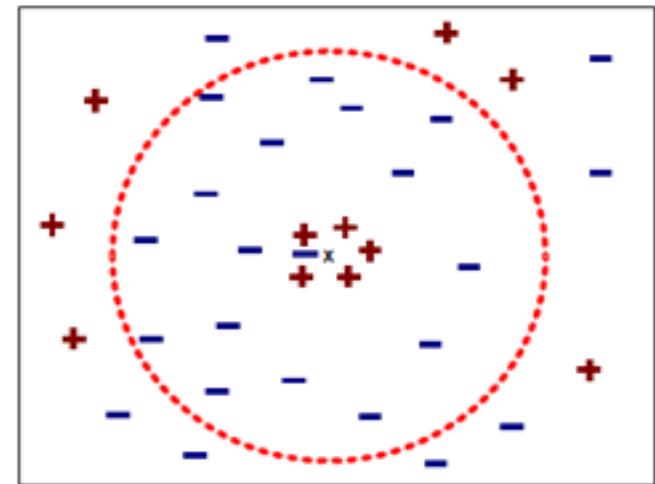
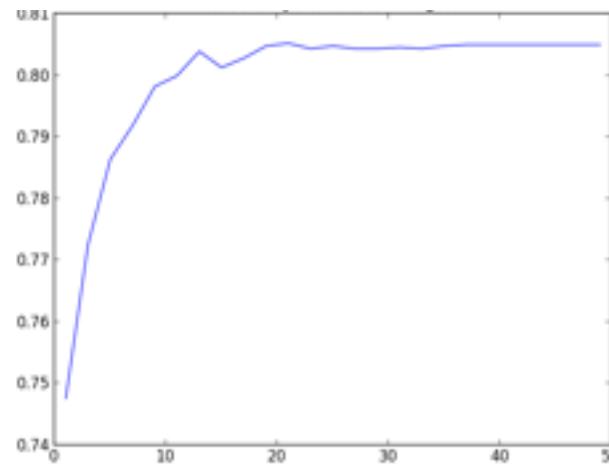
Outros usos: *Imputation* (atribuição)

- Preencher espaços vazios de uma imagem com ruídos



Questão de pesquisa: qual melhor valor de k?

- k muito pequeno → sensibilidade a ruídos
- k muito grande → baixa acurácia
- Na literatura:
 - Medir experimentalmente a partir de k = 1;
 - Valores usuais:
 - 1, 3 e \sqrt{n}



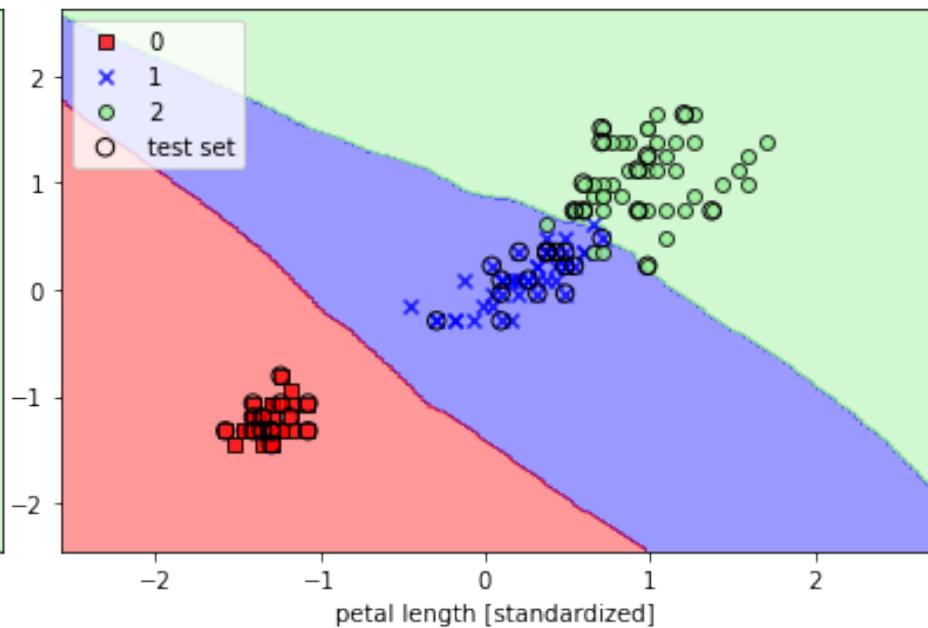
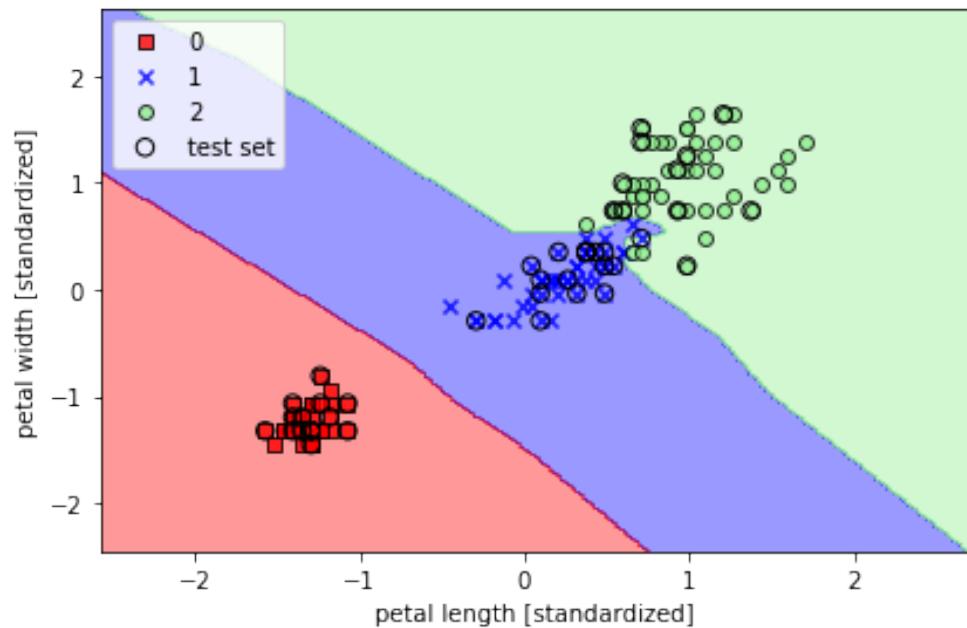
Treinamento kNN

```
1 from sklearn.neighbors import KNeighborsClassifier  
2  
3 knn = KNeighborsClassifier(n_neighbors=10, p=1, metric='minkowski')  
4 knn.fit(X_train_std, y_train)  
5  
6 X_combined_std = np.vstack((X_train_std, X_test_std))  
7 y_combined = np.hstack((y_train, y_test))  
8  
9 plot_decision_regions(X_combined_std, y_combined,  
10                         classifier=knn, test_idx=range(105,150))  
11  
12 plt.xlabel('petal length [standardized]')  
13 plt.ylabel('petal width [standardized]')  
14 plt.legend(loc='upper left')  
15 plt.tight_layout()  
16 # plt.savefig('./figures/k_nearest_neighbors.png', dpi=300)  
17 plt.show()
```

$p = 2$ (dist euclidiana)

$k = 1$

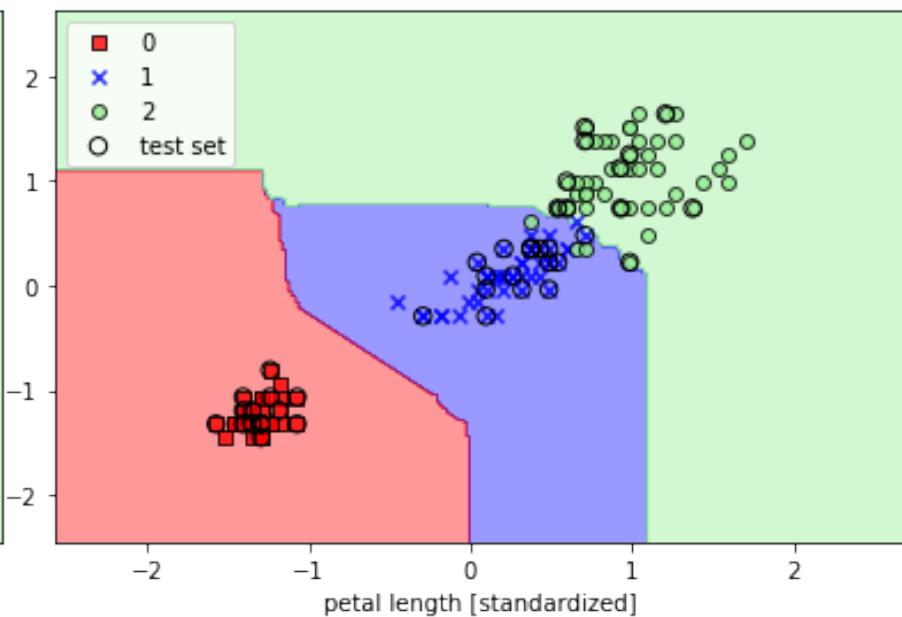
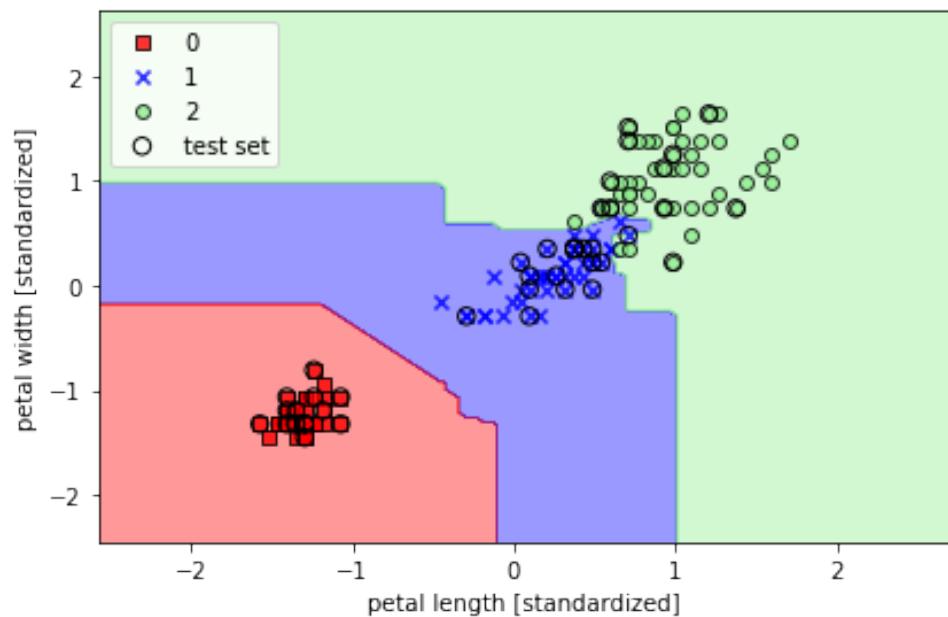
$k = 10$



$p = 1$ (dist manhattan)

$k = 1$

$k = 10$



Divulgação científica



Hendrik Macedo

Escreve sobre Inteligência Artificial no Saense.

<http://www.saense.com.br/autores/artigos-publicados-por-hendrik-macedo/>