ACH2016 - Inteligência Artificial
Aula 07 - k-Vizinhos mais Próximos

Valdinei Freire da Silva

valdinei.freire@usp.br - Bloco A1 100-O

Russell e Norvig, Capítulo 18

Tarefa de Aprendizado Supervisionado

Dado um conjunto de treinamento com N exemplos de pares entrada-saída

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N),$$

onde cada y_i foi gerado por uma função f desconhecida, isto é, $y_i = f(x_i)$.

Descubra uma função h que aproxima a verdadeira função f.

x é a entrada e y é a saída.

x e y pode ser qualquer valor, números ou categorias, x usualmente é um vetor de valores (atributos).

Algoritmos Estudados

- Árvores de Decisão
- Classificador Linear
- Regressão Logística
- Redes Neurais
- k-Vizinhos mais Próximos

Condições para Aprendizado

Suposição estacionária: existe uma distribuição estacionária (população $\mathcal{E}$) sobre os exemplos que permanece constante no tempo.

Essa suposição permite conectar o passado (treinamento) com o futuro (predição).

Cada exemplo é uma variável aleatória E_j para $1 \leq j \leq n$ cujo valor observado $e_j = (x_j, y_j)$ vem de uma distribuição $\Pr(E_j)$.

As variáveis aleatórias $E_1, E_2, \ldots, E_n, E_{n+1}, E_{n+2}, \ldots$ são independentes e identicamente distribuídas.

Inferência Estatística: se $n \to \infty$, então é possível construir um estimador para a distribuição da população $\Pr(E_j)$.

Condições para Aprendizado

Suposição estacionária nem sempre é verdade:

- um especialista pode especificar valores específicos de x_j, sem seguir nenhuma distribuição
- a coleta de amostras pode ser realizada em um contexto diferente do qual os dados serão utilizados
- o próprio método de aprendizado pode escolher os valores de x_j de interesse
- a distribuição pode mudar ao longo do tempo

Melhor Hipótese

Genericamente pode-se pensar em uma função de perda:

$$L(h, \mathcal{E})$$

que avalia a qualidade da hipótese h aplicada na população $\mathcal{E}$.

A hipótese ótima é dada por:

$$h^* = \arg\min_{h \in \mathcal{H}} L(h, \mathcal{E}).$$

Empiricamente, para um conjunto de exemplos E, temos:

$$\hat{h}^* = \arg\min_{h \in \mathcal{H}} L(h, E).$$

Exemplo de função de perda (*Loss Function*):

- Acurácia: taxa de exemplos que são classificados corretamente.
- Verosimilhança: apenas para hipóteses probabilísticas.

Avaliação

Estamos interessado no desempenho da hipótese h em situações ainda não presenciadas, então como avaliá-la?

- testar a hipótese obtida em situações futuras
- testar a hipótese no próprio conjunto de amostras
- testar a hipótese em um subconjunto do conjunto de amostras, mas que não foram utilizadas no treinamento

Avaliação

Validação Cruzada

- holdout: divida o conjunto de amostras aleatoriamente em duas partições, utilize um conjunto para treinar e outro para avaliar
- k-fold: divida o conjunto de amostras aleatoriamente em k partições, utilize uma partição para avaliar e as outras para treinar. Faça um rodízio entre a escolhida para avaliação.
- leave-one-out: utilize uma única amostra para avaliação, e todas as outras amostras para treinar. Faça um rodízio entre a amostra escolhida para avaliação.

Avaliação

Problemas com a Validação Cruzada:

- no holdout como dividir o conjunto: poucas amostras de treinamento, obtém-se um treinamento ruim, poucas amostras para avaliação obtém-se uma avaliação ruim.
- no k-fold deve-se realizar k treinamentos.
- obtém-se uma avaliação do procedimento de treinamento, mas obtém-se várias hipóteses:
  - retreina com todos os exemplos
  - escolhe a hipótese com melhor desempenho

Seleção de Modelo

Estratégia no Aprendizado por Reforço: generalização. O que se aprende com exemplos de treinamento, pode ser generalizado para exemplos na população.

Garantir que ao minimizar $L(h, E)$ é um bom representante para minizar $L(h, \mathcal{E})$.

Overfitting: espaço de hipótese $\mathcal{H}$ pode representar bem os exemplos de treinamento, mas não exemplos novos.

Underfitting: espaço de hipótese $\mathcal{H}$ não pode representar bem os exemplos de treinamento.

Seleção de Modelo

Algoritmo Iterativo:

- considera uma ordenação/partição do espaço de hipóteses de modelos mais simples até modelos mais complexos
- divide o conjunto de amostras em: conjunto de treinamento e conjunto de validação
- iterativamente, enquanto a hipótese aprendida junto ao conjunto de treinamento apresenta melhor avaliação no conjunto de validação, considera modelos cada vez mais complexos
- retorna a hipótese com melhor avaliação no conjunto de validação
- suposição: a avaliação do conjunto de validação tem apenas um ponto de mínimo referente à complexidade

Seleção de Modelo

Regularização:

- considera o seguinte problema de otimização

$$L(h, E) = EmpLoss(h, E) + \lambda Complexity(h)$$

- $EmpLoss(h)$ indica a perda empírica obtida com a hipótese h
- $Complexity(h)$ indica a complexidade da hipótese h
- $\lambda > 0$ que compatibiliza complexidade e perda
- Como escolher λ? teoricamente? empiricamente?

Modelos Não-Paramétricos

Modelos Paramétricos: sumariza os dados em um conjunto de parâmetros de tamanho fixo.

Modelos Não Paramétricos: não pode ser caracterizado por um conjunto finito de parâmetros.

Aprendizado baseado em Instância: armazena algumas (todas no limite) amostras como parte do modelo.

K-vizinhos mais próximos: segundo alguma medida de distância, considera os k vizinhos mais próximos para escolher a saída do sistema.

k-Nearest Neighbors

Qual Distância utilizar?

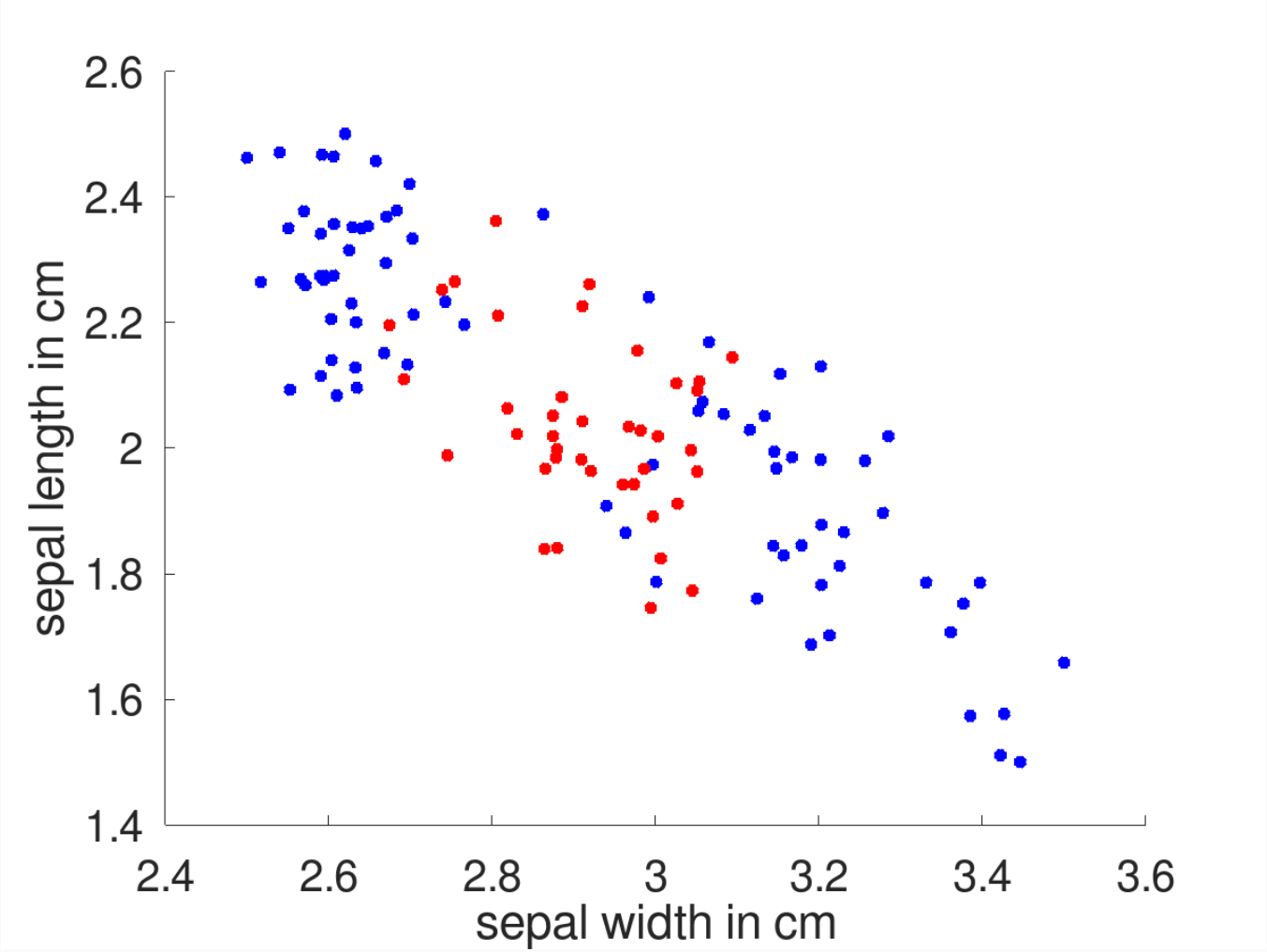
- Distância de Minkowski: $D^{\rho}(\mathrm{x}_p, \mathrm{x}_q) = (\sum_{i=1}^{d} |x_{p,i} - x_{q,i}|^{\rho})^{1/\rho}$, quando $\rho = 2$ tem-se a distância euclidiana, quando $\rho = 1$ tem-se a distância de Manhattan.
- Para evitar problemas com mudança na escala, é comum aplicar normalizações, por exemplo, garantindo que a variância em qualquer dimensão seja 1
- Distância de Mahalanobis: leva em conta a covariância Σ entre as dimensões

$$D^{M}(\mathrm{x}_p, \mathrm{x}_q) = \sqrt{(\mathrm{x}_p - \mathrm{x}_q)^{\top} \Sigma^{-1} (\mathrm{x}_p - \mathrm{x}_q)}$$

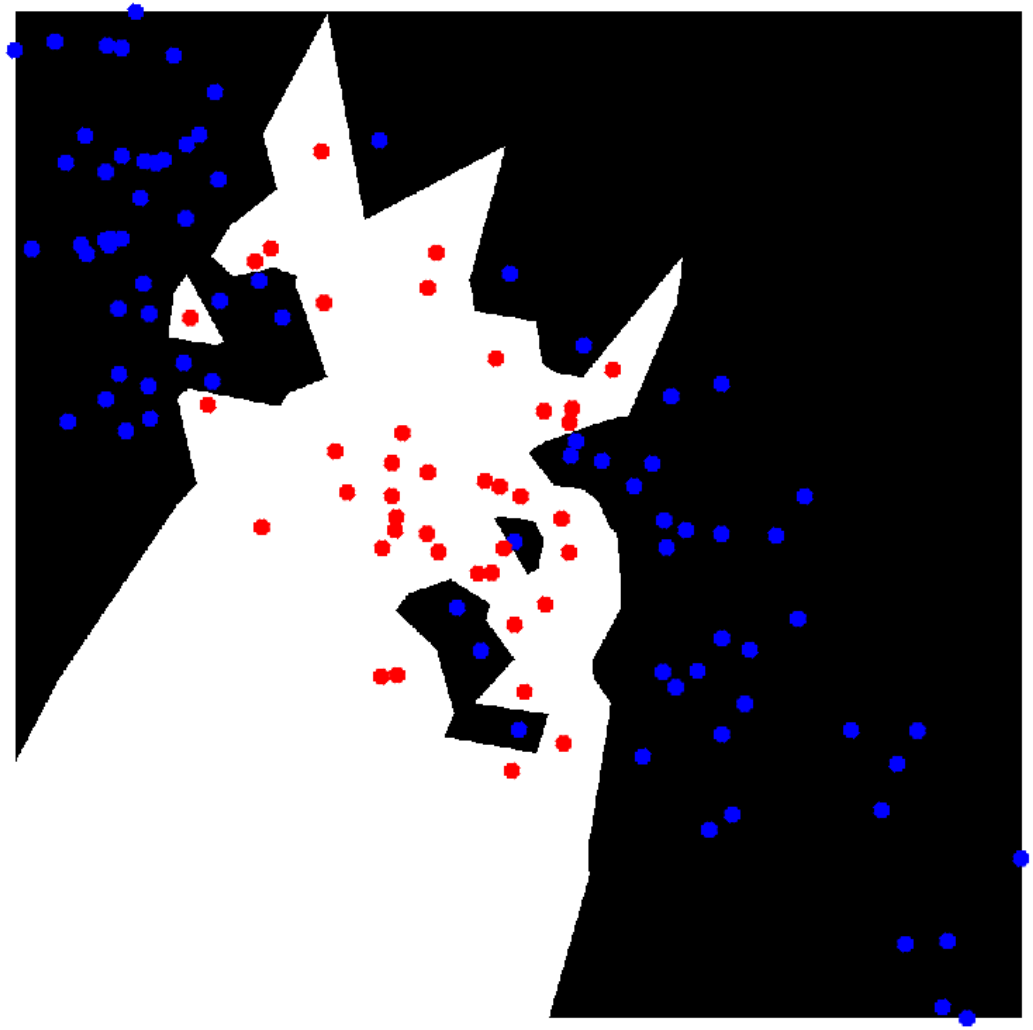
Como escolher a resposta?

- Classificação: realiza votação (ponderada) entre os k vizinhos mais próximos

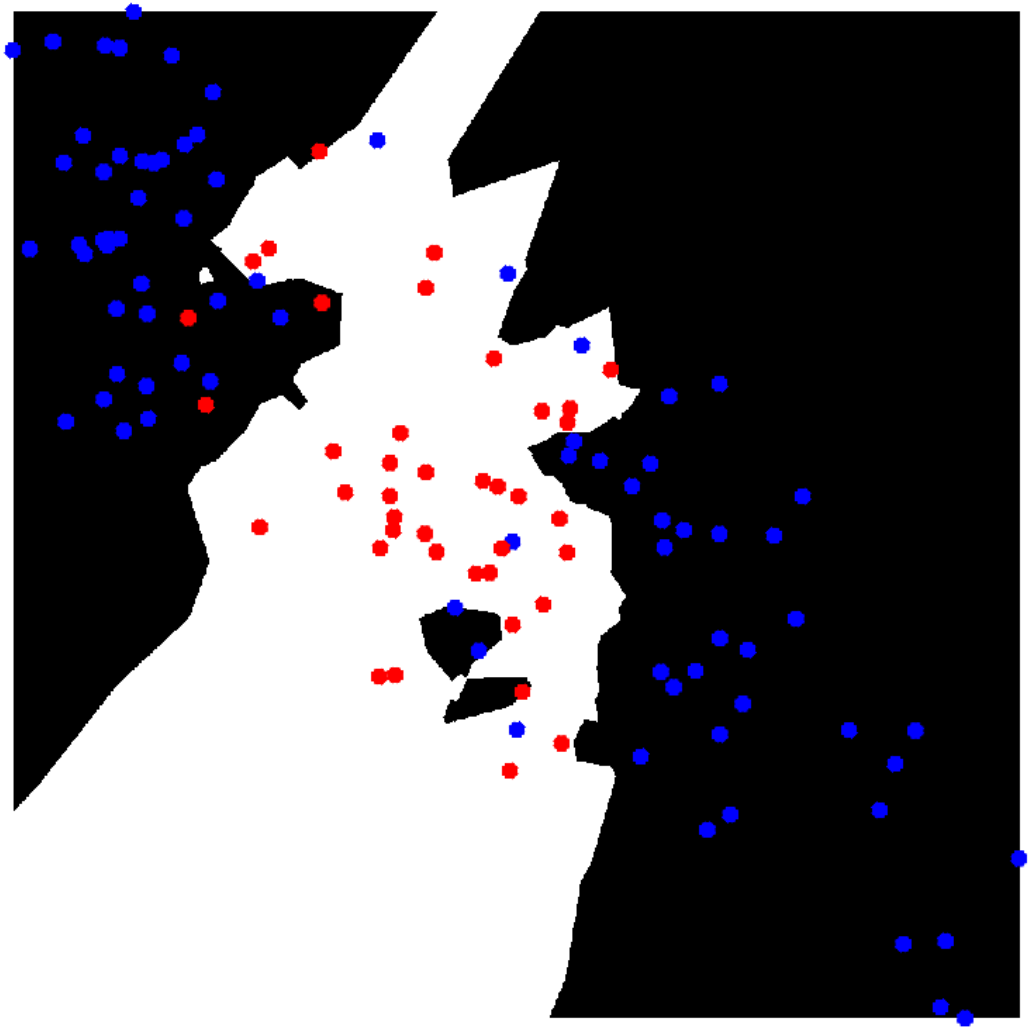
Exemplos


2.6
2.4
2.2
2
1.8
1.6
1.4
sepal length in cm
2.4
2.6
2.8
3
3.2
3.4
3.6
sepal width in cm

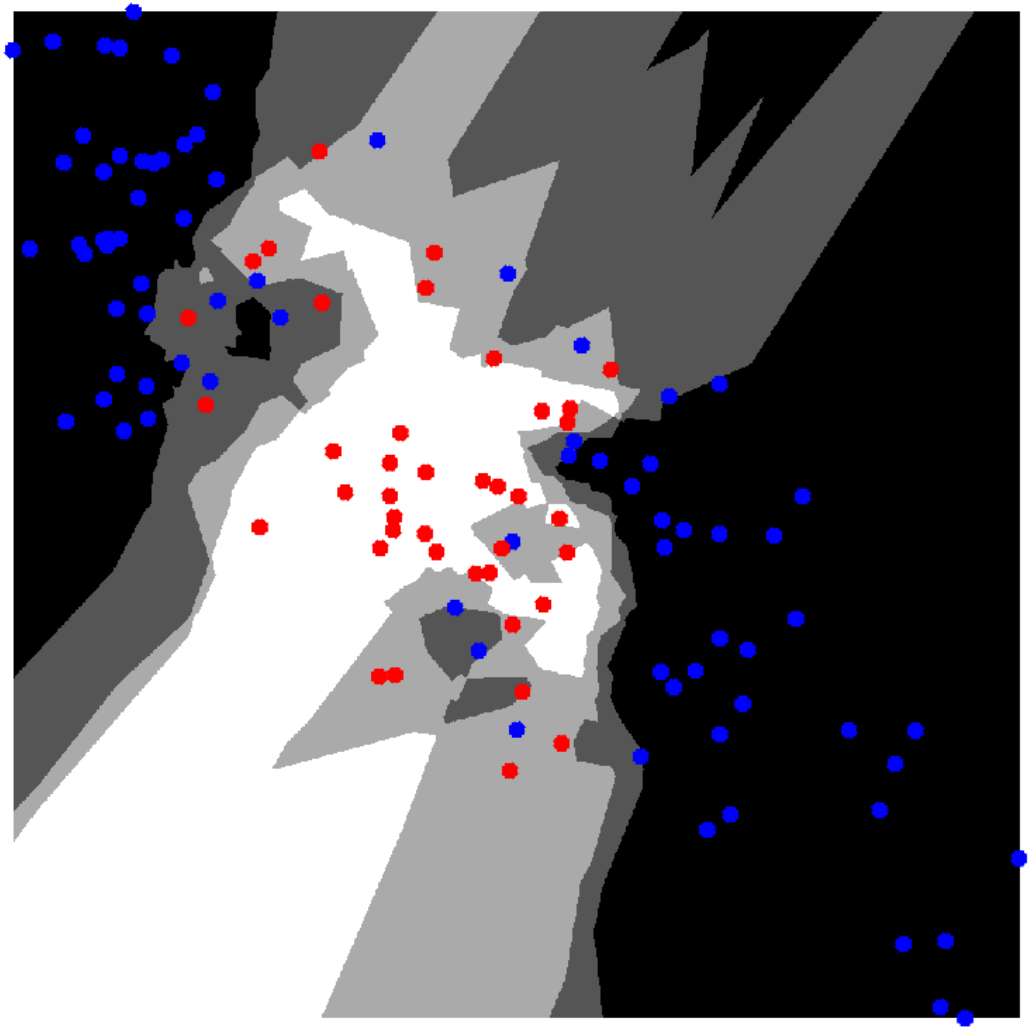
k-NN (k=1)



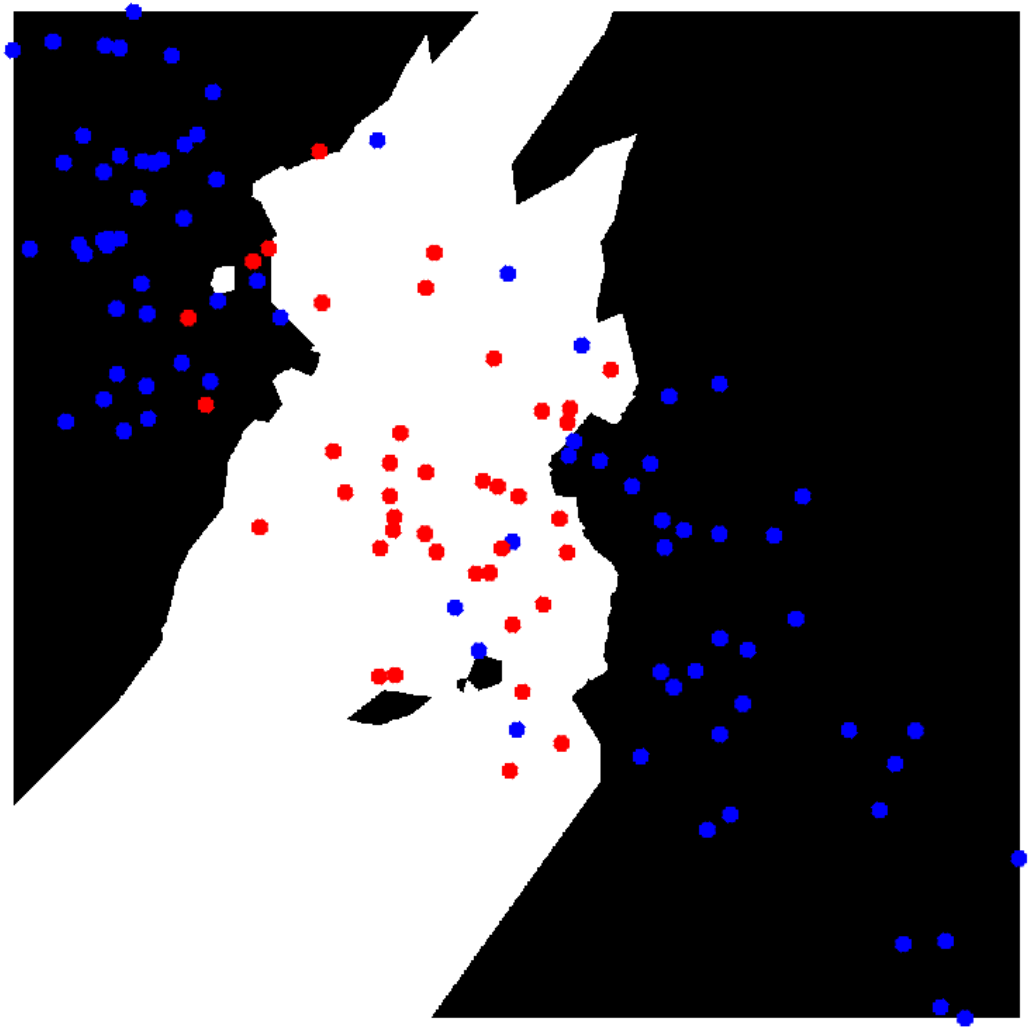
k-NN (k=3)



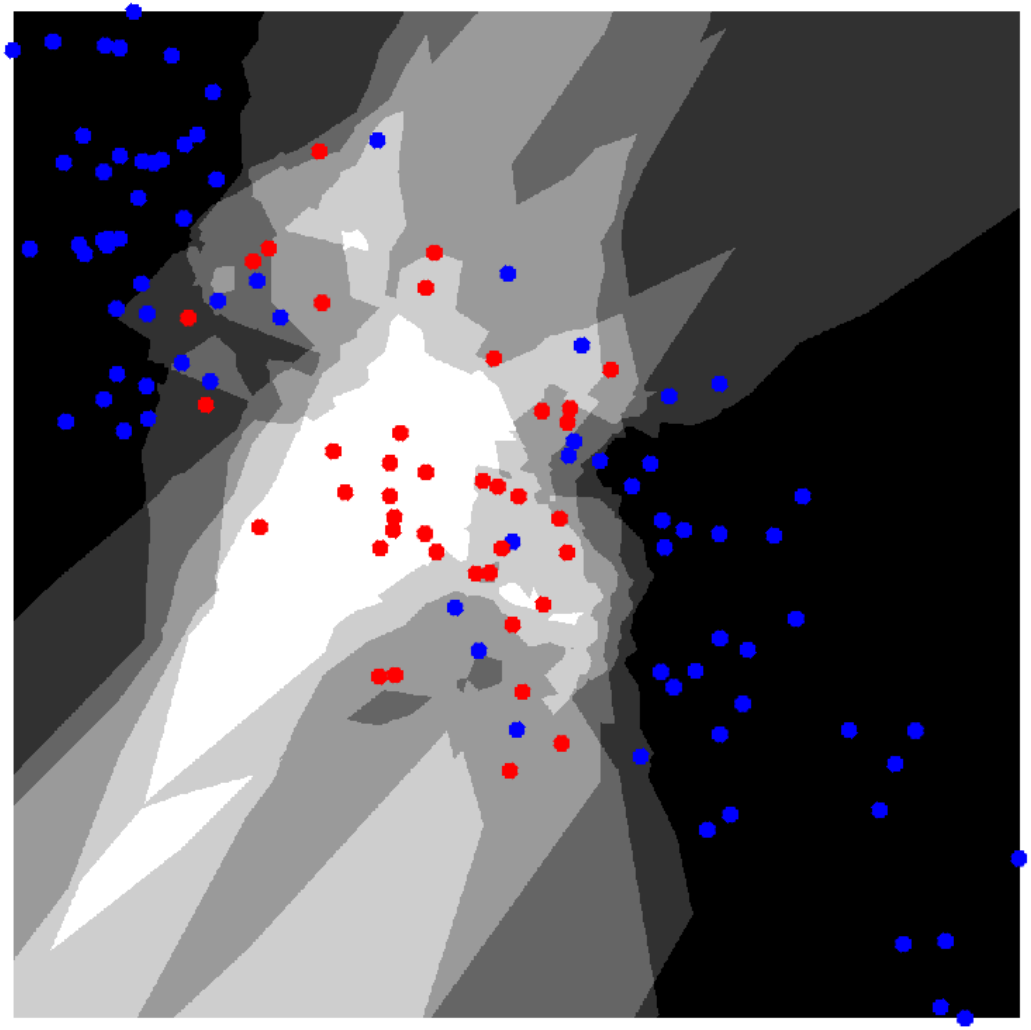
k-NN (k=3)



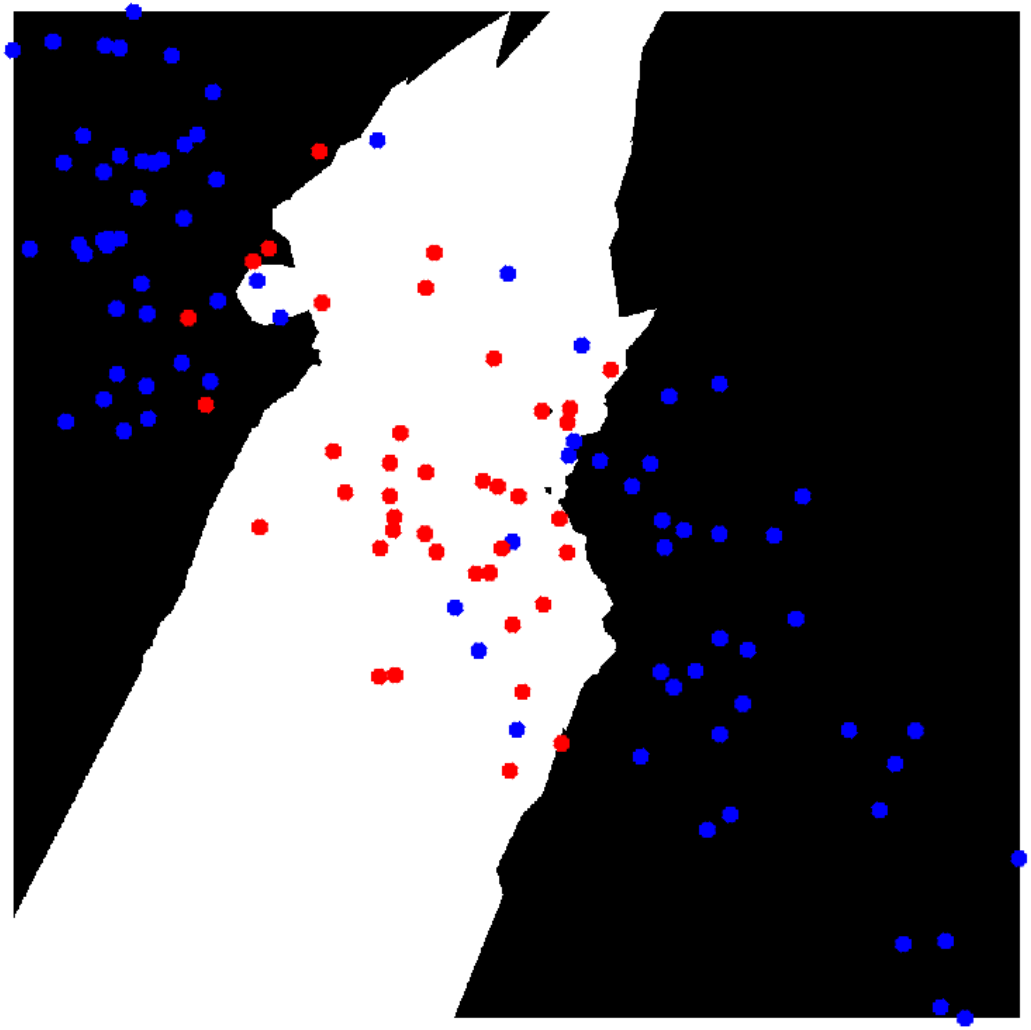
k-NN (k=5)



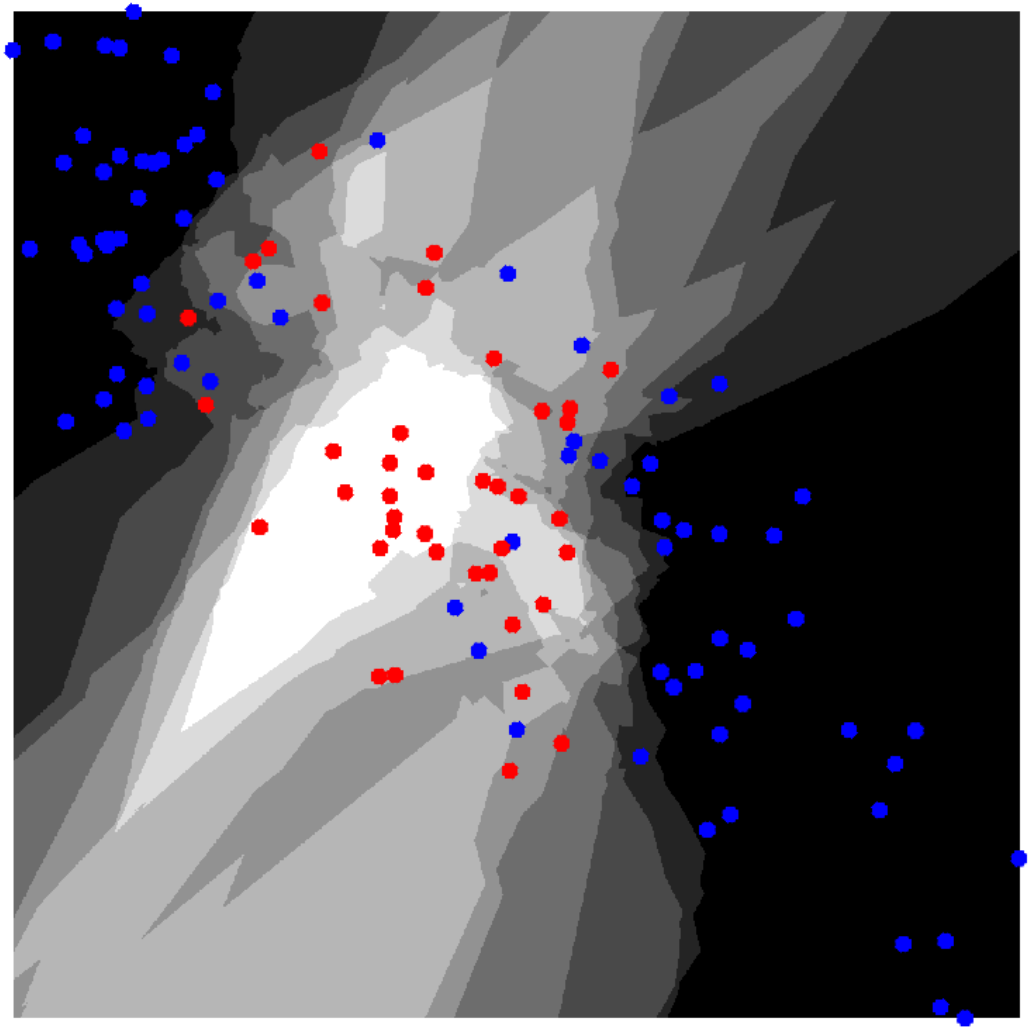
k-NN (k=5)



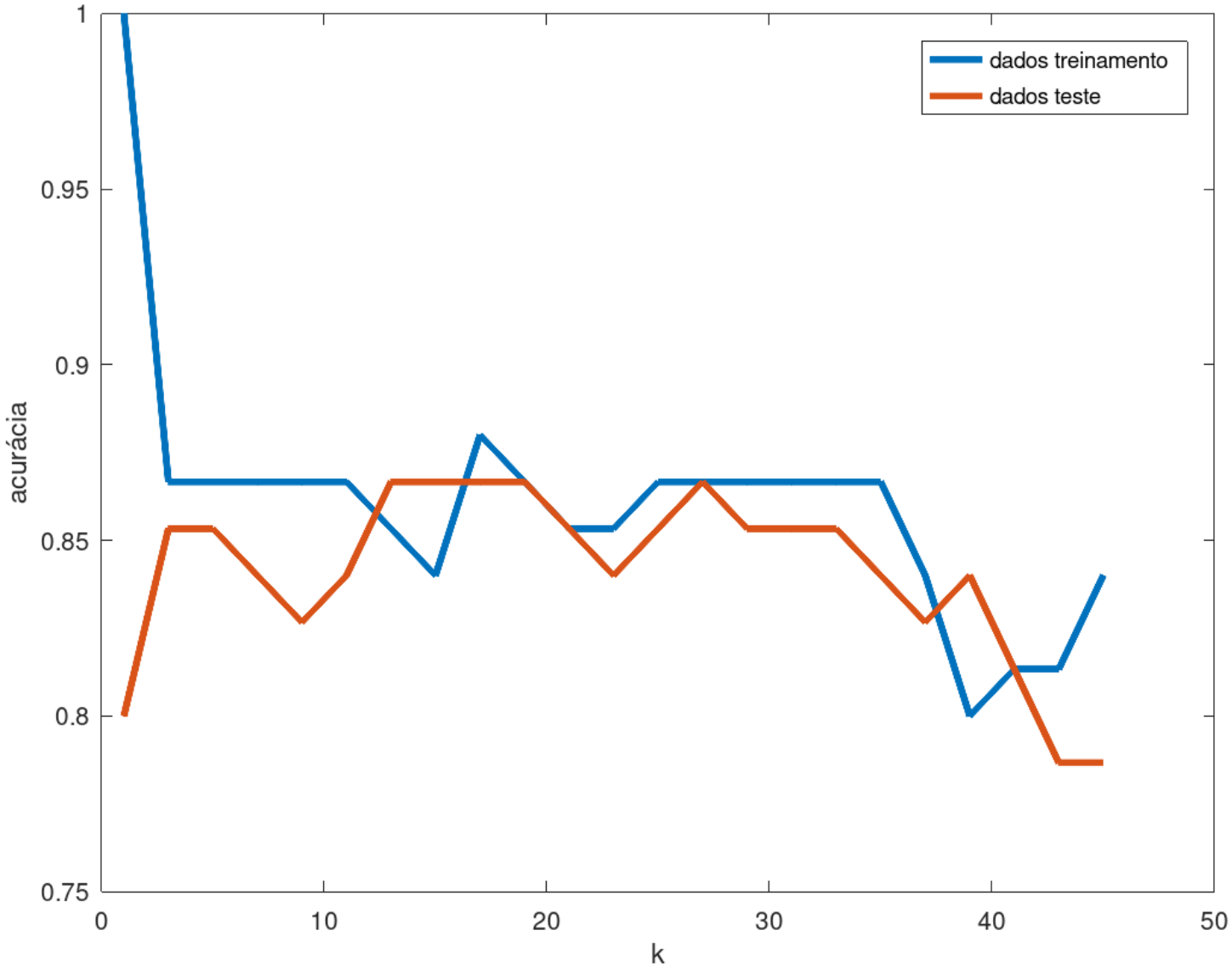
k-NN (k=7)



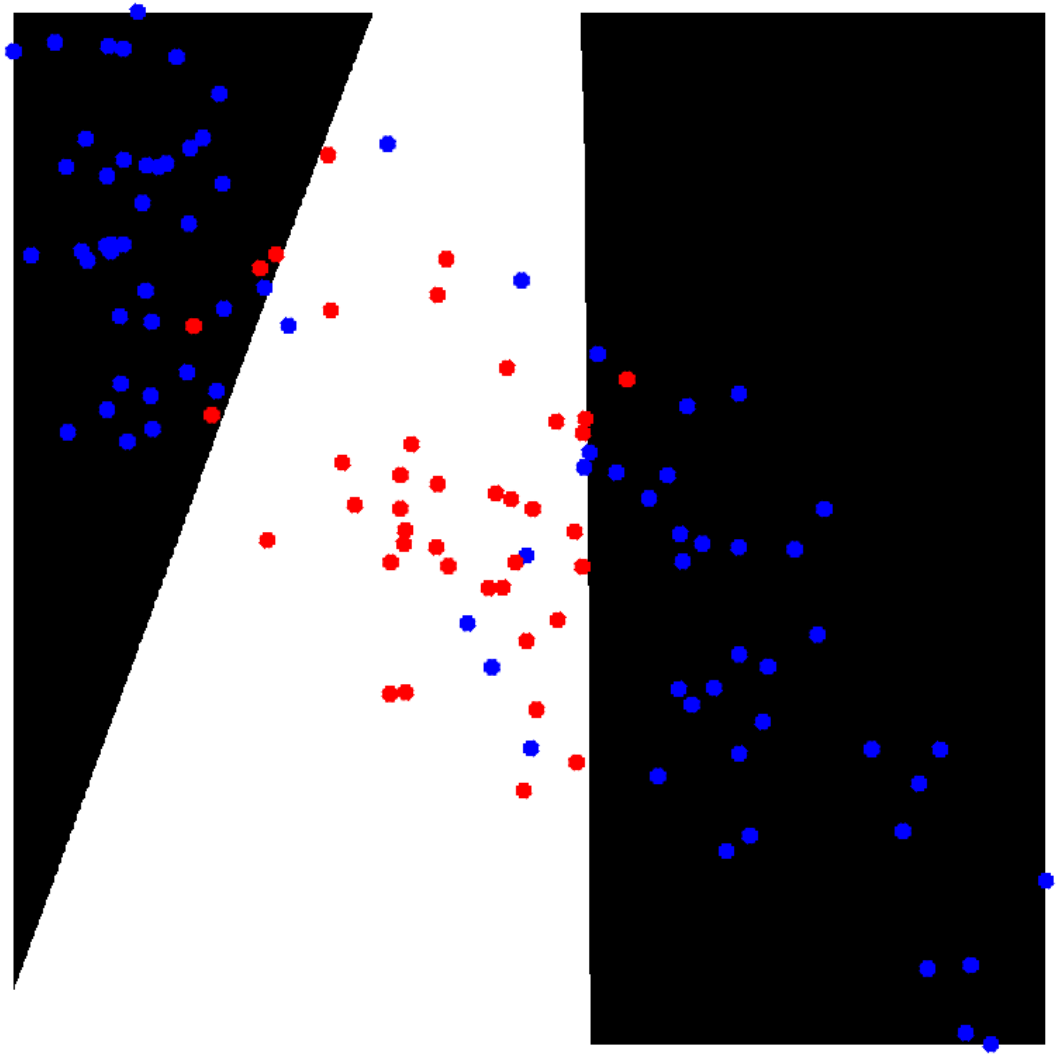
k-NN (k=7)



Escolha de Modelo



Redes Neurais (2 Neurônios)



Redes Neurais (10 Neurônios)