

## 03. Algoritmos Supervisionados

*Modelos não lineares*

---

ISEC, IC 22-23

CPereira

1

### Naive Bayes

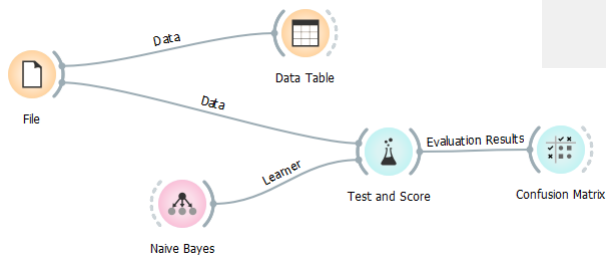
- Princípios

- Semelhantes aos modelos lineares apresentados no capítulo anterior, mas geralmente tendem a ser ainda mais rápidos no processo de treino.
- **Aprendizagem** baseia-se na análise individual de cada atributo a recolha de estatísticas para cada classe. Exemplos:
  - BernoulliNB - recolhe com que frequência cada atributo de cada classe é diferente de zero.
  - MultinomialNB - leva em consideração o valor médio de cada atributo para cada classe.
  - GaussianNB - armazena o valor médio, bem como o desvio padrão de cada atributo para cada classe.
- Para realizar uma **Predição**:
  - O novo exemplo é comparado com as estatísticas de cada uma das classes e a melhor classe correspondente é atribuída.

2

# Naive Bayes

- ...
- exemplo

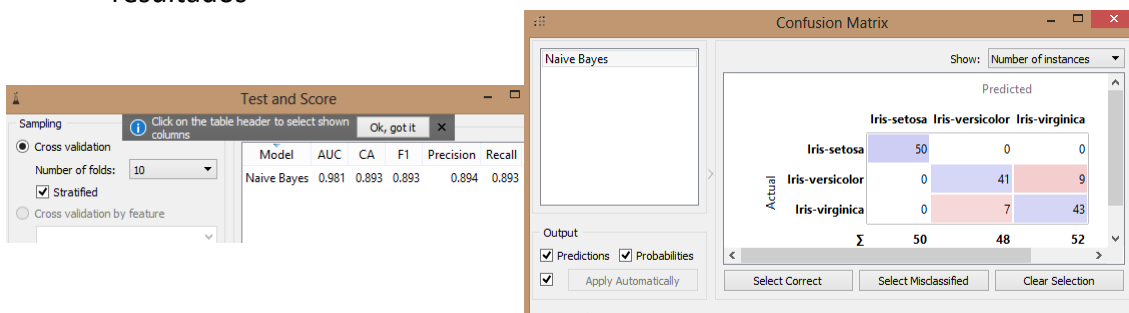


	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.1	3.5	1.4	0.2
2	Iris-setosa	4.9	3.0	1.4	0.2
3	Iris-setosa	4.7	3.2	1.3	0.2
4	Iris-setosa	4.6	3.1	1.5	0.2
5	Iris-setosa	5.0	3.6	1.4	0.2
6	Iris-setosa	5.4	3.9	1.7	0.4
7	Iris-setosa	4.6	3.4	1.4	0.3
8	Iris-setosa	5.0	3.4	1.5	0.2
9	Iris-setosa	4.4	2.9	1.4	0.2
10	Iris-setosa	4.9	3.1	1.5	0.1
11	Iris-setosa	5.4	3.7	1.5	0.2
12	Iris-setosa	4.8	3.4	1.6	0.2
13	Iris-setosa	4.8	3.0	1.4	0.1
14	Iris-setosa	4.3	3.0	1.1	0.1
15	Iris-setosa	5.6	4.0	1.2	0.2
16	Iris-setosa	5.7	4.4	1.5	0.4
17	Iris-setosa	5.4	3.9	1.3	0.4

3

# Naive Bayes

- ....
- resultados



4

# Árvores de decisão

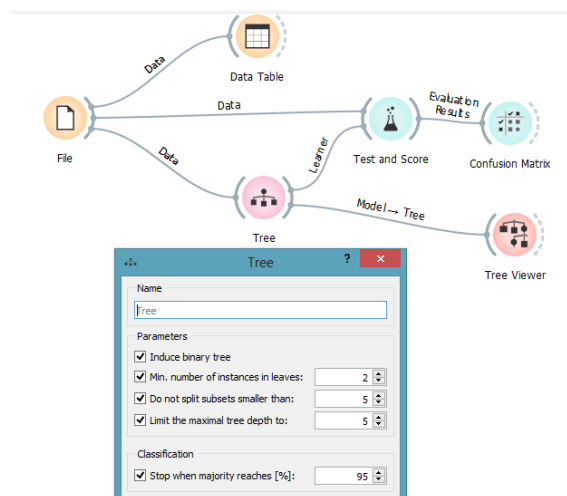
- Árvores de decisão

- são modelos amplamente usados para tarefas de classificação e regressão.
- Essencialmente, aprendem uma hierarquia de perguntas se/senão, levando a uma decisão.
- Para construir uma árvore, o algoritmo pesquisa todos os testes possíveis e encontra aquele que é mais informativo sobre a variável de destino.
- Controlo de complexidade
  - Normalmente, construir uma árvore até que todas as folhas sejam puras leva a modelos que são muito complexos e altamente ajustados aos dados de treino - A presença de folhas puras significa que uma árvore é 100% precisa no conjunto de treino;
  - Deve-se assim limitar a profundidade da árvore.

5

# Árvores de decisão

- ...
- Iris dataset

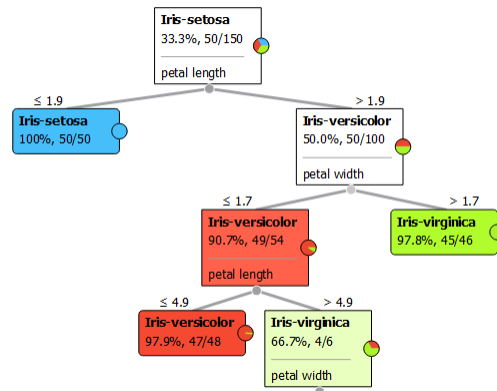


6

# Árvores de decisão

• ...

		Predicted		
		Iris-setosa	Iris-versicolor	Iris-virginica
Actual	Iris-setosa	50	0	0
	Iris-versicolor	0	47	3
	Iris-virginica	0	4	46
$\Sigma$		50	51	49



7

## Random forests

### • Princípios

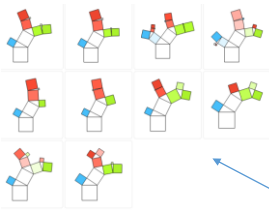
- Pretendem colmatar uma das principais desvantagens das árvores de decisão, o sobre-ajustamento aos dados de treino (overfitting);
- Consistem simplesmente numa coleção de árvores de decisão, onde cada árvore é ligeiramente diferente das outras.
  - cada árvore pode realizar um trabalho relativamente bom de previsão, mas provavelmente ajustará demasiado parte dos dados de treino.
  - Injetam aleatoriedade na construção da árvore, para garantir que cada árvore seja diferente.
  - Se construirmos muitas árvores, podemos reduzir o overfitting e manter a capacidade de generalização, calculando a saída como a média de resultados.

8

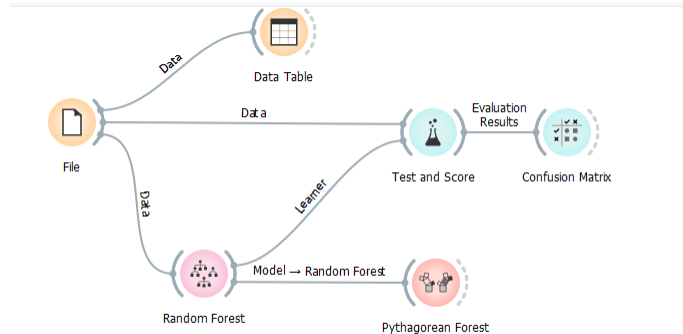
# Random forests

• ...

- Iris dataset



- Árvores que representam nós com quadrados, cujo tamanho é proporcional ao número de instâncias de treino abrangidas.

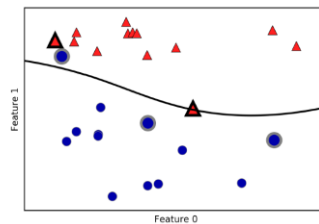


9

# Máquinas de Vetores de Suporte

- “Kernelized Support Vector Machines”

- As máquinas de vetor de suporte com kernel (geralmente chamadas apenas de SVMs) são uma extensão para problemas de classificação não linear.
  - Baseiam-se no princípio de que adicionar atributos não lineares à representação de nossos dados pode tornar os modelos mais poderosos. Contudo estes atributos são considerados de forma implícita através de uma função não linear (kernel);
  - Normalmente, apenas um subconjunto dos exemplos de treino importam para definir o limite de decisão: aqueles que ficam na fronteira entre as classes – designados de vetores de suporte.

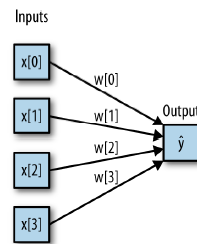


10

# Redes Neurais

## • Arquitetura

- Vamos considerar por agora modelos mais simples de redes com várias camadas.
- Podemos partir do conceito da regressão logística e de modelos lineares e generalizar o conceito.
  - Modelo linear.



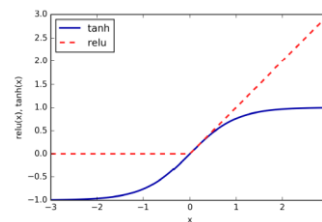
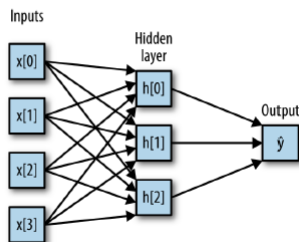
- Existem vantagens em adicionar mais camadas a um modelo linear?

11

# Redes Neurais

## • ...

- Considerando mais uma camada e funções de ativação não lineares, poderemos resolver problemas mais complexos...

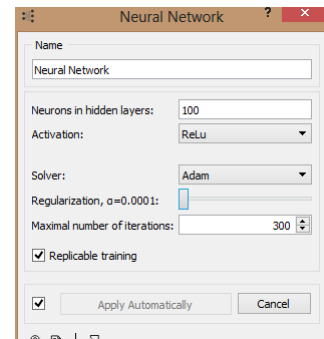
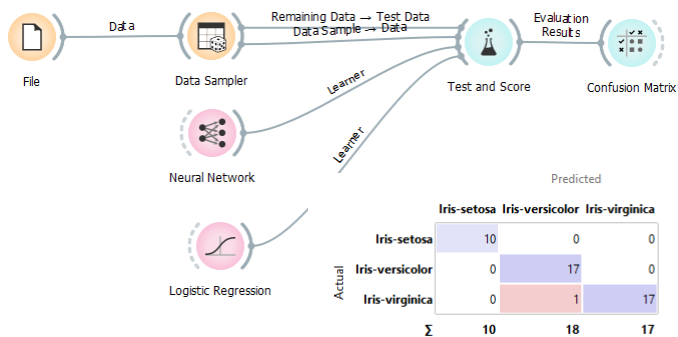


12

# Redes Neurais

• ...

## • Exemplo – Iris



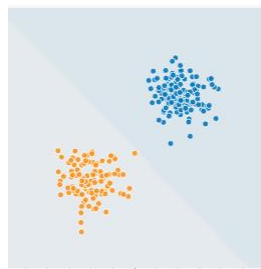
- Para este problema, justifica-se a aplicação de uma rede neuronal?

13

# Redes Neurais

• ...

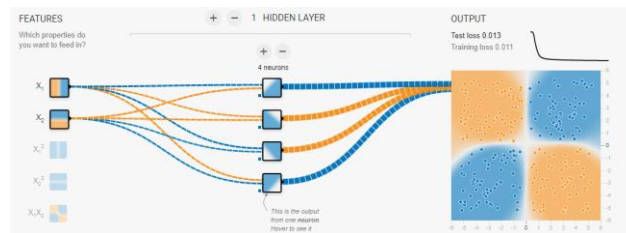
- Para problemas muito complexos, necessitamos de muitas camadas e parâmetros – “deep learning”
- <https://playground.tensorflow.org/>
  - Defina uma arquitetura MLP para o seguinte problema de classificação. Avalie redes simples primeiro!



14

# Redes Neurais

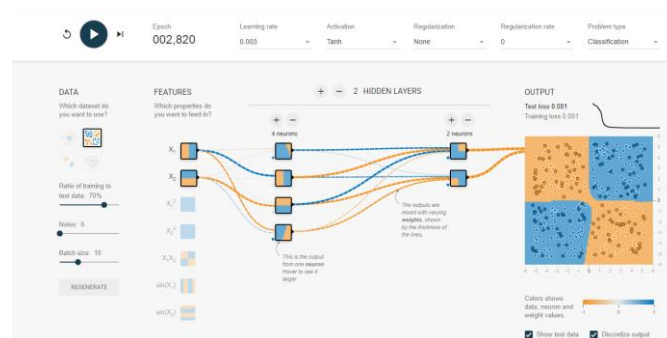
- ...
  - Problema não linearmente separável
    - Uma camada com 2 neurónios e função de ativação não linear - Será suficiente?
    - Uma camada interna com 4 neurónios:



15

# Redes Neurais

- ...
  - 2 camadas internas



16



# Redes Neuronais

- ...
  - Defina uma rede neuronal MLP para o seguinte problema de classificação:



17

## Análise dos Algoritmos

- Qual o modelo a usar?
  - Modelos lineares
    - Deve ser o primeiro algoritmo a tentar!
    - Devido à sua simplicidade ajusta-se a problemas com muitos exemplos e atributos.
  - Naive Bayes
    - Apenas para problemas de classificação
    - Rápido e adequado a datasets de elevada dimensão. Usualmente é menos preciso que modelos lineares.

18

# Análise

- ...
  - Árvores de decisão
    - Tempos de treino reduzidos
    - Interpretabilidade – fáceis de visualizar e explicar!
  - Random forests
    - Melhoram o desempenho das árvores de decisão.
    - Robustas e precisas. Não necessitam de normalização do dados
    - Pouco adequadas para datasets com muitos atributos (problemas de dimensão elevada) ou de informação esparsa.

19

# Análise

- ...
  - Máquinas de vetores de suporte
    - Adequam-se a problemas de dimensão média com características de significado similar.
    - Necessitam de normalização dos dados.
    - Sensíveis aos parâmetros.
  - Redes neuronais
    - Permitem a definição de modelos muito complexos, particularmente para datasets de elevada dimensão.
    - Necessitam habitualmente de normalização dos dados e sensíveis aos parâmetros de treino.
    - Tempos de treino muito elevados para problemas complexos.

20