02. Algoritmos Supervisionados

Generalização
Modelos lineares
K-nearest neighbours

ISEC, IC 22-23
CPereira

1

Aprendizagem

- Os tipos de aprendizagem automática são habitualmente classificados de acordo com:
  - O facto de serem ou não treinados com supervisão humana
    - **Supervisionada**
    - Não supervisionada
    - Semi-supervisionada
    - Aprendizagem por reforço
  - Se eles podem ou não aprender de forma incremental em tempo real (aprendizagem online ou em *batch* - lote).
  - Se simplesmente comparam novos exemplos (de teste) com exemplos já conhecidos (de treino) ou, em vez disso, detetam padrões nos dados de treino para construir um modelo preditivo (baseado em instância versus aprendizagem baseada em modelos)

2

Algoritmos de Aprendizagem Supervisionada

- Aprendizagem supervisionada
  - Usada sempre que queremos prever um determinado resultado de um determinado exemplo de teste e temos exemplos observados de pares de entrada/saída (treino).
  - Construímos um **modelo de aprendizagem** a partir desses pares de entrada/saída, que compõem o conjunto de treino.
  - Objetivo: **realizar previsões** precisas para exemplos novos, nunca antes vistos.
    - O conjunto de teste nunca deve ser usado para adaptar qualquer parâmetro do modelo.
    - Requer ainda o esforço humano para construir o conjunto de treino, mas depois automatiza a tarefa de predição.

3

Algoritmos

- Tipos principais de problemas de aprendizagem supervisionada:
  - **Classificação**
    - Prever um rótulo de classe, que consiste numa escolha a partir de uma lista predefinida;
    - Exemplo: Classificação da Iris dataset
      - **3 classes** pré definidas (virginica, setosa, versicolor) ; 4 atributos (largura da sépala, comprimento da sépala, largura da pétala, comprimento da pétala)
      - Outros exemplos? "Wine classification", "Spam classification", "Mnist" – classificação de dígitos manuscritos….
  - **Regressão**
    - Prever um número contínuo ou real.
    - Exemplo: Prever o valor de um série temporal (Ficha 1 das aulas práticas) é um exemplo de uma tarefa de regressão.
      - Outros exemplos? Predição de um índice numa bolsa de valores ("stock market"), …
    - A forma de distinguir entre tarefas de classificação e regressão é avaliar se existe algum tipo de continuidade na variável de saída a prever.

4

Conceito de Generalização

- Como **avaliar o sucesso** de aprendizagem?
  - Fase I - Construir um modelo com base nos dados de treino,
  - Fase II - Fazer previsões "precisas" sobre exemplos novos e nunca observados, que tenham as mesmas características (atributos ou "features") do conjunto de treino.

  Se um modelo é capaz de fazer previsões precisas sobre dados ainda não vistos, dizemos que ele é capaz de **generalizar** do conjunto de treino para o conjunto de teste.
  - Queremos um modelo que seja capaz de generalizar com a maior precisão possível.
    - Normalmente, construímos um modelo capaz de realizar previsões precisas sobre o conjunto de treino.
    - Se os conjuntos de treino e teste têm muitas características em comum, esperamos que modelo também seja preciso no conjunto de teste.
    - No entanto, existem alguns casos em que isto não acontece - por exemplo, se construirmos modelos muito complexos, demasiado "ajustado" aos dados de treino (é sempre possível ser tão preciso quanto quisermos no conjunto de treino!!).

5

Generalização

- Exemplo
  - 12 amostras (treino)
  - 6 atributos
  - Duas classes
  - Problema?
    - Irá um novo cliente comprar um barco?

Age	Number of cars owned	Owns house	Number of children	Marital status	Owns a dog	Bought a boat
66	1	yes	2	widowed	no	yes
52	2	yes	3	married	no	yes
22	0	no	0	married	yes	no
25	1	no	1	single	no	no
44	0	no	2	divorced	yes	no
39	1	yes	2	married	yes	no
26	1	no	2	single	no	no
40	3	yes	1	married	yes	no
53	2	yes	2	divorced	no	yes
64	2	yes	3	divorced	no	no
58	2	yes	2	married	yes	yes
33	1	no	1	single	no	no

- *Fonte: Introduction to Machine Learning with Python by Andreas C. Muller and Sarah Guido*

6

Generalização

- …
  - Modelo baseado em regras
    - "Se cliente tem mais de 45 anos e (menos de 3 filhos ou é não divorciado)

      Então: compra um barco"
    - Como avaliar este modelo?
      - É 100% preciso para o conjunto de treino!!
      - Mas poderá generalizar corretamente para novos exemplos?
        - Parece demasiado complexo e suporta-se num conjunto reduzido – por exemplo "não divorciado" suporta-se apenas num exemplo!
      - A única medida para saber se um algoritmo terá um bom desempenho é a avaliação no conjunto de teste.
      - Intuitivamente esperamos que modelos simples generalizem melhor para novos dados. E se a regra fosse "Pessoas com mais de 50 anos querem comprar um barco"?

7

Generalização

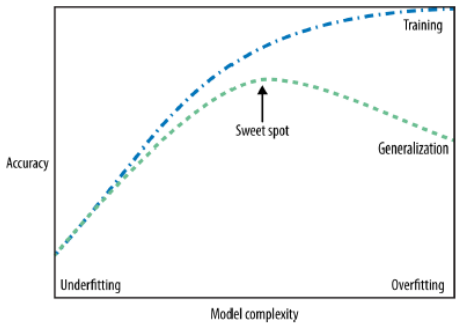
- Overfitting
  - Ocorre quando se ajusta um modelo muito próximo às particularidades do conjunto de treino
    - neste caso obtém-se um modelo que funciona bem no conjunto de treino, mas não é capaz de generalizar para novos exemplos.

- Underfitting
  - E se o modelo for muito simples, "O cliente que possui um casa compra um barco "
    - Neste caso, não captura todos os aspetos e variabilidade dos dados
    - O modelo tem um mau desempenho mesmo no conjunto de treino.

8

Generalização

- …
  - Devemos encontrar um ponto de "trade-off" – compromisso:



- *Fonte: Introduction to Machine Learning with Python by Andreas C. Muller and Sarah Guido*

Generalização

- …
  - Complexidade do modelo vs Dimensão do conjunto de treino
    - Quanto maior a variedade de dados no conjunto de treino, mais complexo poderá ser o modelo a obter, sem overfitting.
      - Normalmente, a recolha de mais exemplos (sem duplicar…) gera maior variedade. Assim, conjuntos de treino maiores permitem a construção de modelos mais complexos.
    - **Perícia humana** - decidir quantos dados recolher e tratar - pode ser mais benéfico do que ajustar simplesmente o modelo!

Algoritmos mais comuns

- Para além de dados "artificiais", vamos considerar dois casos de estudo reais de pequena dimensão:
  - Para classificação
    - "Wisconsin Breast Cancer" dataset
      - 569 exemplos, com 30 "features".
  - Para regressão:
    - Boston Housing dataset
      - 506 exemplos descritos por 13 "features".
  - https://github.com/rupakc/UCI-Data-Analysis

11

Modelos Lineares

- Os modelos lineares realizam uma previsão com base num modelo (função) <u>linear</u> das varáveis de entrada (*features*):
  - Para **<u>regressão</u>**, a fórmula para um modelo linear é a seguinte:

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \ldots + w[p] * x[p] + b$$

    - *Entrada (*x*); p Features* x[i] i=0,…p *; saída =* ŷ
  - O modelo consiste numa linha reta para uma única "feature", um plano ao usar duas "features" ou um híper-plano para dimensões superiores.
  - Existem muitas variantes de algoritmos lineares para regressão. A diferença entre estes modelos reside em como os parâmetros (w e b) são aprendidos a partir dos dados de treino – <u>algoritmo</u>.

12

Modelos Lineares

- **Linear regression (least squares)**
  - A regressão linear, ou mínimos quadráticos (LS), é o método linear mais simples e clássico para problemas de regressão.
    - encontra os parâmetros w e b que minimizam o erro quadrático médio entre as previsões ($\mathbf{y}_{modelo}$) e os alvos de regressão ($\mathbf{y}_{target}$)
      - O erro quadrático médio é calculado pela soma das diferenças quadradas entre as previsões e os valores verdadeiros.
  - Estes modelos de regressão linear não possuem híper-parâmetros para ajustar, o que é um benefício, mas por outro lado não existe forma de controlar a complexidade do modelo.

13

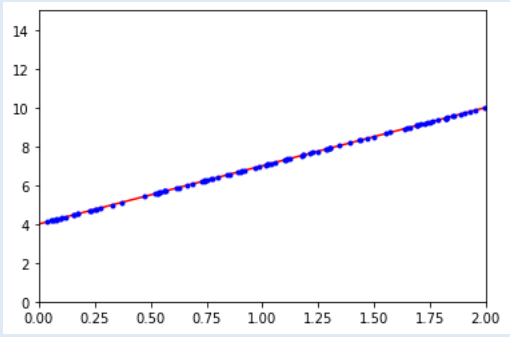
Regressão Linear

- …
  - exemplo

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt

# dataset
X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X

#plot
plt.plot(X, y, "b.")
plt.axis([0, 2, 0, 15])
plt.show()
```



14

Regressão Linear

- …
  - Equação normal - minimiza MSE
    y = theta*X
    theta = inv (X'*X) * X' * y (X' – transposta de X)
  - code:

```
X_b = np.c_[np.ones((100, 1)), X]  # adiciona x0 = 1 a cada instância

theta_best = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
print(theta_best)
```

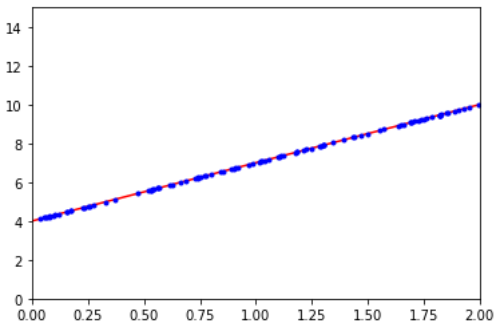
15

Regressão Linear

- …
  - Avaliação no conjunto de teste:

```
X_new = np.array([[0], [2]])
X_new_b = np.c_[np.ones((2, 1)), X_new] # add x0 = 1 to each instance
y_predict = X_new_b.dot(theta_best)

plt.plot(X_new, y_predict, "r-")
plt.plot(X, y, "b.")
plt.axis([0, 2, 0, 15])
plt.show()
```



16

Regressão Linear

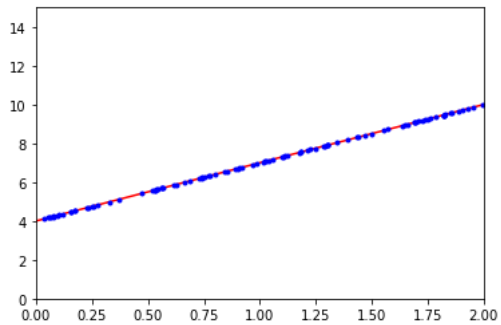
- ...
  - Implementação no sk-learn

```
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X, y)

print(lin_reg.coef_)
print(lin_reg.intercept_)

y_predict=lin_reg.predict(X_new)

plt.plot(X_new, y_predict, "r-")
plt.plot(X, y, "b.")
plt.axis([0, 2, 0, 15])
```



17

Regressão Linear

- ...
  - Conjunto de dados mais complexo,
    - Boston Housing.
    - 506 amostras e 105 atributos.
      - Usar “LinearRegression” de sklearn
      - https://github.com/rupakc/UCI-Data-Analysis/blob/master/Boston%20Housing%20Dataset/boston.py

18

Ridge Regression

- Outro modelo linear para regressão,
  - Mesma fórmula que mínimos quadrados normal, porém, os coeficientes (w) são escolhidos não apenas para obter uma boa predição mas também para se ajustar uma restrição adicional:
    - a magnitude dos coeficientes deve ser o menor possível - devem estar perto de zero.
    - Intuitivamente, isso significa que cada atributo deve ter um efeito no resultado final tão pequeno quanto possível.
    - Esta restrição é um exemplo do que é chamado de **regularização** - restringindo um modelo para evitar overfitting.
      - conhecida como regularização L2.

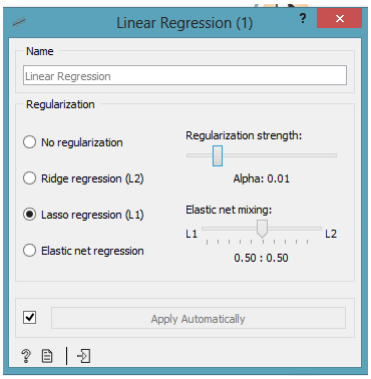
Lasso Regression

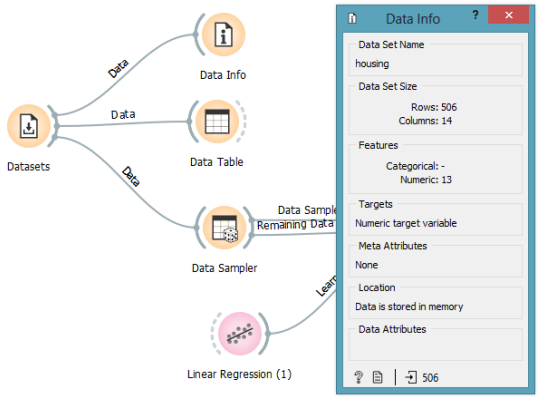
- Faz seleção de atributos
  - restringe os coeficientes para serem próximos de zero, mas alguns coeficientes são exatamente zero.
    - alguns atributos são totalmente ignorados pelo modelo.
    - Torna o modelo mais fácil de interpretar e pode revelar as características mais importantes dos dados.

  - Exercício
    - Aplicar estas formas de regressão ao “Boston Housing”
    - Quais os atributos mais relevantes?

Lasso Regression

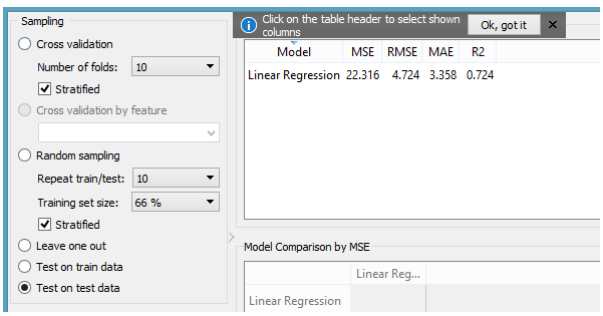
- Exemplo
  - Boston housing dataset





Lasso Regression

- …
  - RMSE
    - root mean squared error
  - MSE
    - mean squared error
  - MAE
    - mean absolute error
  - R-quadrado score
    - O R-quadrado é uma medida estatística de quão próximos os dados estão da linha de regressão ajustada.
    - entre 0 e 100%. quanto maior o R-quadrado, melhor o modelo se ajusta aos seus dados.



Modelos Lineares para Classificação

- Os modelos lineares também são amplamente usados para classificação.
  - Para classificação binária, o modelo é representado por:

    $\hat{y} = w[0] * x[0] + w[1] * x[1] + ... + w[p] * x[p] + b > 0$

    - A fórmula é muito semelhante à da regressão linear. Se a função for menor que zero, predizemos a classe –1; se for maior que zero, nós prever a classe +1.

  - Existem muitas formas diferentes de encontrar os coeficientes. Os dois algoritmos mais comuns são:

    Regressão logística,
    - implementado em "linear_model.LogisticRegression"

    Máquinas de vetores de suporte lineares
    - implementado em "svm.LinearSVC" (SVC – suport vector classification).

    - Apesar do nome, "LogisticRegression" é um algoritmo apenas de **classificação** e não de regressão, e não deve ser confundido com "LinearRegression".

23

Modelos Lineares para Classificação

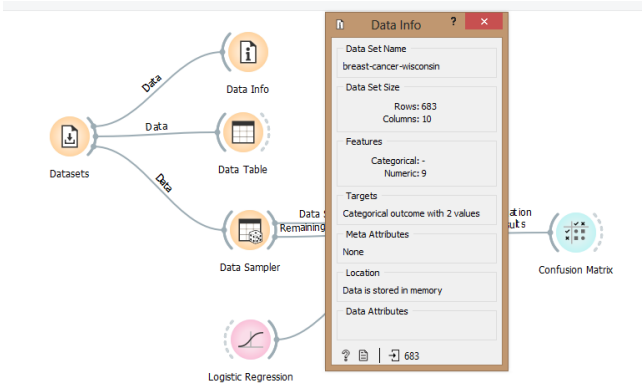
- To do work

  - Aplicar ao problema "Breast Cancer dataset" o seguintes modelos:
    - LinearLogistic
    - SCV

24

Modelos Lineares para Classificação

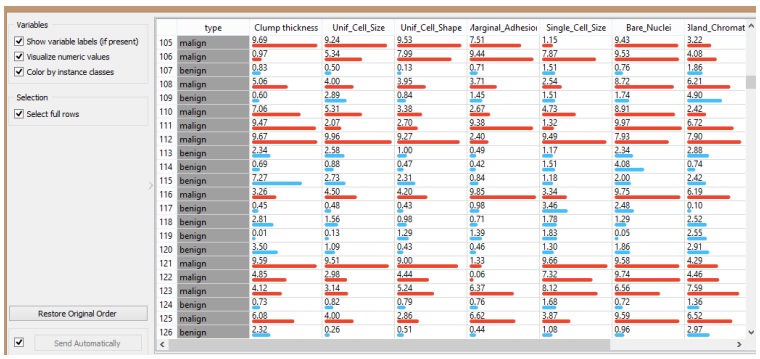
- Linear Logistics



25

Modelos Lineares para Classificação

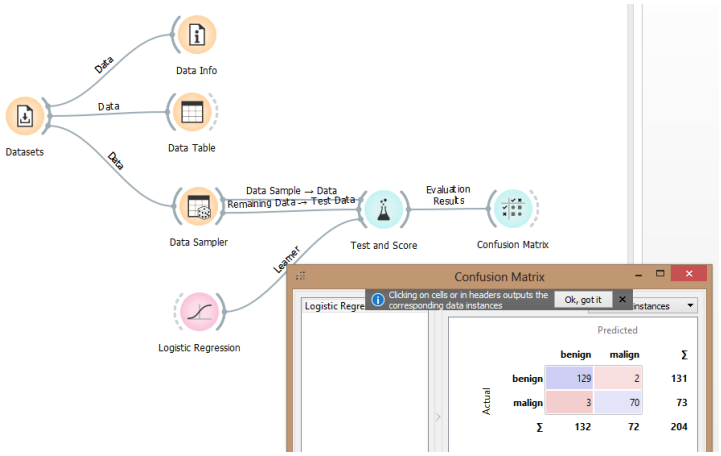
- …
  - Data Table



26

Modelos Lineares para Classificação

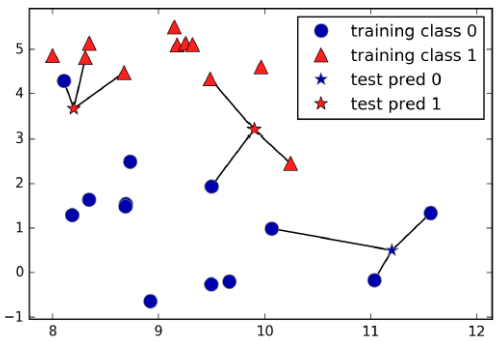
- …
  - Matriz de confusão



27

k-Nearest Neighbors

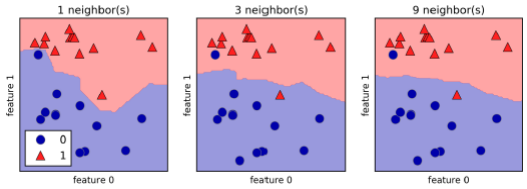
- Os k-vizinhos mais próximos – "instance based learning"
  - O algoritmo mais simples. Consiste apenas em armazenar o conjunto de dados de treino ("espécie" de modelo).
  - Para efetuar uma predição, o algoritmo encontra os exemplos mais próximos (k).
  - A predição é então a saída/classe conhecida para esses pontos de treino - a classe maioritária entre os k-vizinhos mais próximos.



28

k-Nearest Neighbors

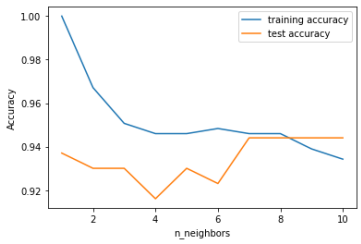
- Análise de desempenho:
  - utilizar um **único vizinho** resulta numa curva de decisão limite que segue de perto os dados de treino.
  - Considerar mais vizinhos leva a uma fronteira de decisão mais suave. Um limite mais suave corresponde a um modelo mais simples.
    - No caso extremo, em que o número de vizinhos é igual ao número de exemplos de treino, todas as previsões seriam iguais à classe mais frequente!



29

k-Nearest Neighbors

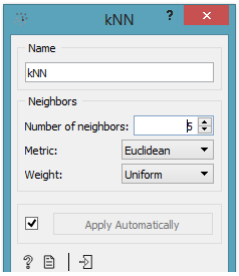
- Exemplo:
  - To do work:
    - investigar se podemos confirmar a conexão entre a complexidade do modelo e a generalização em dois conjuntos de dados:
      - "Iris"
      - "Breast Cancer Dataset"

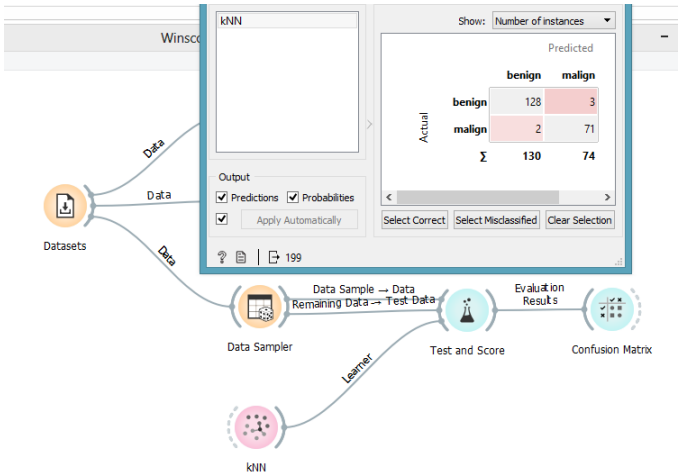


30

k-Nearest Neighbors

- …
  - Winsconsin dataset





31

k-Nearest Neighbors

- Análise
  - Pontos fortes:
    - Interpretabilidade do modelo (fácil de entender).
    - usar um pequeno número de vizinhos (três a cinco) geralmente funciona bem!
  - Pontos fracos:
    - Quando o conjunto de treino é muito grande (em número de amostras ou atributos), a previsão pode ser bastante demorada.
    - Não costuma ser usado na prática devido a esta dificuldade em lidar com dados reais de larga escala.

32

Referências

- *Introduction to Machine Learning with Python by Andreas C. Muller and Sarah Guido;*
- *https://github.com/rupakc/UCI-Data-Analysis*
- *http://orange.biolab.si/docs/*

33