

# Mineração de Dados

## Regressão



- 1 Conceitos Básicos
- 2 Vizinhos mais Próximos
- 3 Método dos Mínimos Quadrados
- 4 Regressão Ridge
- 5 Regressão Lasso
- 6 Avaliação de Modelos
- 7 Outros Métodos e Contextos

# Conceitos Básicos

# Aprendizado Supervisionado e Não Supervisionado

- ▶ Aprendizado Supervisionado
  - ▶ Classificação e regressão
  - ▶ Supervisão: os dados de treinamento são acompanhados por valores esperados
  - ▶ Os valores esperados podem vir de observações, medições, indicações de um especialista, etc
  - ▶ Novas instâncias são classificadas com o que se aprendeu sobre os dados de treinamento

# Regressão e Classificação

- ▶ **Regressão**

- ▶ Predição numérica
- ▶ Os modelos são funções que retornam valores contínuos

- ▶ **Classificação**

- ▶ O modelo deve prever uma classe para um conjunto de valores de entrada

# Vizinhos mais Próximos

# Vizinhos mais Próximos

- ▶ Os algoritmos dos Vizinhos mais Próximos podem usados em
  - ▶ Aprendizado não supervisionado
  - ▶ Aprendizado supervisionado: regressão ou classificação
- ▶ <http://scikit-learn.org/stable/modules/neighbors.html>
- ▶ O algoritmo dos  $k$  Vizinhos mais Próximos ( $k$ NN) requer uma definição de vizinhança e o número de vizinhos
  - ▶ Várias métricas de distância estão disponíveis<sup>1</sup>
  - ▶ A vizinhança pode ser guardada em estruturas de dados mais adequadas para consulta
  - ▶ O número de vizinhos pode ser alterado para um limiar de distância máxima

---

<sup>1</sup> <http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html>

# Vizinhos mais Próximos

- ▶ Os algoritmos dos Vizinhos mais Próximos podem usados em
  - ▶ Aprendizado não supervisionado
  - ▶ Aprendizado supervisionado: regressão ou classificação
- ▶ <http://scikit-learn.org/stable/modules/neighbors.html>
- ▶ O algoritmo dos  $k$  Vizinhos mais Próximos ( $k$ NN) requer uma definição de vizinhança e o número de vizinhos
  - ▶ Várias métricas de distância estão disponíveis<sup>1</sup>
  - ▶ A vizinhança pode ser guardada em estruturas de dados mais adequadas para consulta
  - ▶ O número de vizinhos pode ser alterado para um limiar de distância máxima

---

<sup>1</sup> <http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html>



# Buscando os Vizinhos mais Próximos

- ▶ Dada uma nova instância
- ▶ Montar uma matriz de distâncias
- ▶ Ordenar os elementos de acordo com a distância
- ▶ Retornar os  $k$  vizinhos mais próximos
- ▶ A ordenação dos elementos pode ser substituída
  - ▶ Manter a lista de vizinhos ordenada
  - ▶ Diminui o custo computacional quando há uma grande quantidade de instâncias e poucos vizinhos são requeridos
- ▶ A ordenação dos elementos também não é necessária quando os vizinhos são definidos por um limiar de distância

# Buscando os Vizinhos mais Próximos

- ▶ Dada uma nova instância
- ▶ Montar uma matriz de distâncias
- ▶ Ordenar os elementos de acordo com a distância
- ▶ Retornar os  $k$  vizinhos mais próximos
- ▶ A ordenação dos elementos pode ser substituída
  - ▶ Manter a lista de vizinhos ordenada
  - ▶ Diminui o custo computacional quando há uma grande quantidade de instâncias e poucos vizinhos são requeridos
- ▶ A ordenação dos elementos também não é necessária quando os vizinhos são definidos por um limiar de distância

# Buscando os Vizinhos mais Próximos

- ▶ Dada uma nova instância
- ▶ Montar uma matriz de distâncias
- ▶ Ordenar os elementos de acordo com a distância
- ▶ Retornar os  $k$  vizinhos mais próximos
- ▶ A ordenação dos elementos pode ser substituída
  - ▶ Manter a lista de vizinhos ordenada
  - ▶ Diminui o custo computacional quando há uma grande quantidade de instâncias e poucos vizinhos são requeridos
- ▶ A ordenação dos elementos também não é necessária quando os vizinhos são definidos por um limiar de distância

# Buscando os Vizinhos mais Próximos

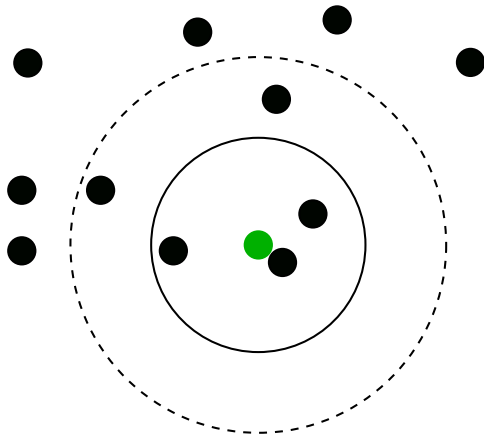


Figura adaptada de

[https://en.wikipedia.org/wiki/K-nearest\\_neighbors\\_algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm)

## $k$ NN: Regressão

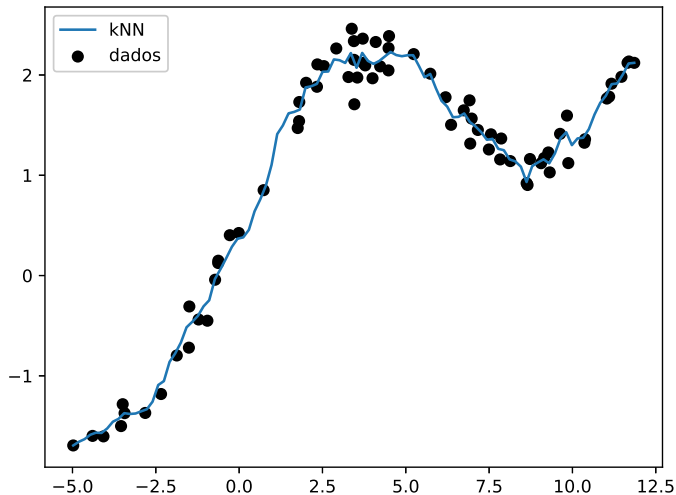
- ▶ Dados os vizinhos mais próximos, pode-se usá-los para calcular uma aproximação para o valor esperado ( $y$ )
  - ▶ O  $y$  foi usado no exemplo anterior apenas para ilustrar a busca pelos vizinhos
  - ▶ Ao resolver o problema de determinar  $f(x_i) \approx y_i$ , os valores de  $x$  (variáveis independentes) são usados nos cálculos de distância
- ▶ Dada uma métrica de distância  $\text{dist}$ , a aproximação é calculada como a média dos valores de  $y$  dos vizinhos

$$y(x) \approx \frac{1}{k} \sum_{i=1}^k y_i \quad y(x) \approx \frac{1}{k} \sum_{i=1}^k w(x, x_i) y_i$$

$$w(x, x_i) = 1 - \frac{\text{dist}(x, x_i)}{\max_{j,l}(\text{dist}(x_j, x_l))} \quad w(x, x_i) = \frac{\exp(-\text{dist}(x, x_i))}{\sum_{j=1}^k \exp(-\text{dist}(x, x_j))}$$

- ▶ Scikit Learn disponibiliza o inverso da distância como ponderação

# $k$ NN: Regressão com Limiar de Distância

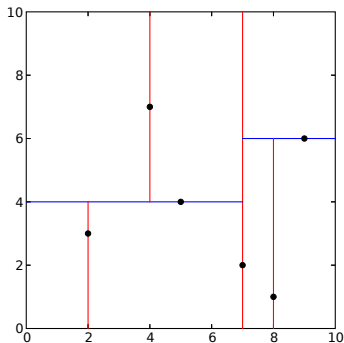
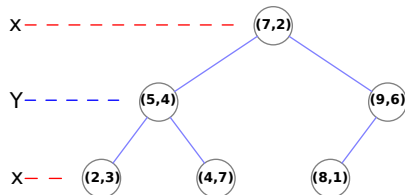


## Vizinhos mais Próximos e *KDTree*

- ▶ A consulta pelos vizinhos mais próximos é computacionalmente caro
- ▶ A avaliação de novas instâncias pode ser proibitiva para grandes quantidades de dados
- ▶ Outras formas de armazenamento podem diminuir o custo computacional das buscas por vizinhos
- ▶ A Scikit Learn possui duas alternativas
  - ▶ KDTree: `algorithm = 'kd_tree'`
  - ▶ BallTree: `algorithm = 'ball_tree'`

# Vizinhos mais Próximos e *KDTree*

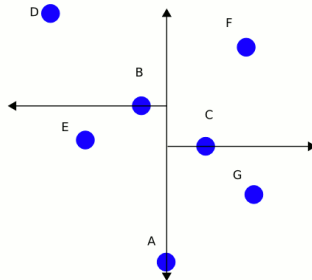
Dados:  $(2,3)$ ,  $(5,4)$ ,  $(9,6)$ ,  $(4,7)$ ,  $(8,1)$ ,  $(7,2)$ .



[http://en.wikipedia.org/wiki/K-d\\_tree](http://en.wikipedia.org/wiki/K-d_tree)



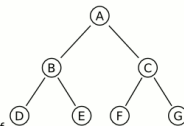
# Vizinhos mais Próximos e *KDTree*



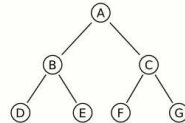
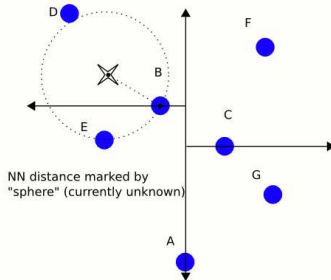
X-Splitting planes

Y-Splitting planes

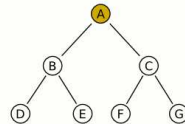
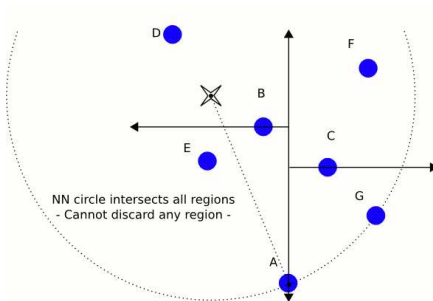
X-Splitting planes  
not needed for leaf



# Vizinhos mais Próximos e *KDTree*

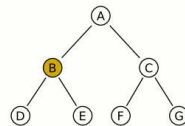
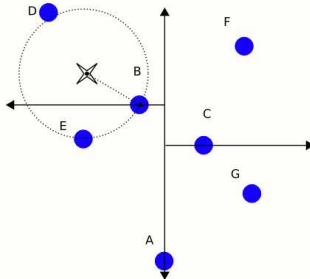


# Vizinhos mais Próximos e *KDTree*



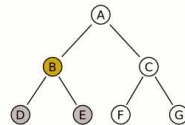
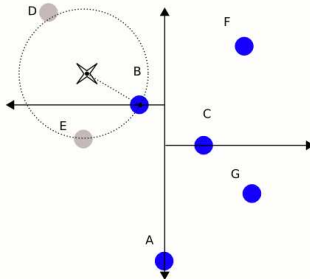
Start at A, then proceed in depth-first search (maintain a stack of parent-nodes if using a singly-linked tree). Set best estimate to A's distance. Then examine left child node

# Vizinhos mais Próximos e *KDTree*



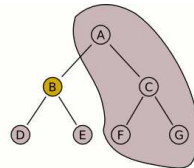
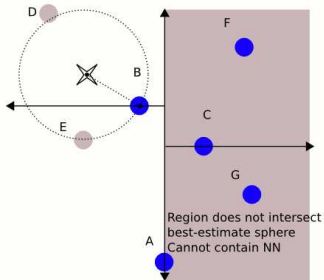
Calculate B's distance and compare against best estimate  
 - It is smaller distance, so update best estimate. Examine children (left then right)

# Vizinhos mais Próximos e *KDTree*



D & E Discarded as B  
(already visited) is closer.  
B is the best estimate for B's sub-branch  
Proceed back to parent node

# Vizinhos mais Próximos e *KDTree*



A's children have all been searched,  
B is the best estimate for entire tree

# Método dos Mínimos Quadrados

## Caso Discreto – Formulação Geral

- ▶ No caso geral, dados  $x_1, \dots, x_m$  e seus correspondentes  $y_1, \dots, y_m$ , deseja-se determinar

$$g(x) = c_0\phi_0(x) + c_1\phi_1(x) + \dots + c_n\phi_n(x) = \sum_{j=0}^n c_j\phi_j(x)$$

- ▶ que minimiza

$$\begin{aligned} E(c_0, c_1, \dots, c_n) &= \langle \mathbf{r}; \mathbf{r} \rangle = \sum_{i=1}^m [y_i - g(x_i)]^2 \\ &= \sum_{i=1}^m \left[ y_i - \sum_{j=0}^n c_j\phi_j(x_i) \right]^2 \end{aligned}$$

- ▶ onde  $\phi_j(x)$  são funções conhecidas e chamadas de funções de base



## Caso Discreto – Formulação Geral

- ▶ Para determinar os parâmetros  $c_0, c_1, \dots, c_n$  que minimizam  $E(c_0, c_1, \dots, c_n)$ , as derivadas parciais com relação a cada um dos parâmetros  $c_k$ ,  $k = 0, \dots, n$ , são igualadas a zero
- ▶ Logo,

$$\frac{\partial E(c_0, \dots, c_n)}{\partial c_k} = 0$$
$$\frac{\partial}{\partial c_k} \left[ \sum_{i=1}^m \left( y_i - \sum_{j=0}^n c_j \phi_j(x_i) \right)^2 \right] = 0$$
$$2 \left[ \sum_{i=1}^m \left( y_i - \sum_{j=0}^n c_j \phi_j(x_i) \right) (-\phi_k(x_i)) \right] = 0$$

## Caso Discreto – Formulação Geral

- ▶ Para determinar os parâmetros  $c_0, c_1, \dots, c_n$  que minimizam  $E(c_0, c_1, \dots, c_n)$ , as derivadas parciais com relação a cada um dos parâmetros  $c_k$ ,  $k = 0, \dots, n$ , são igualadas a zero
- ▶ Logo,

$$\frac{\partial E(c_0, \dots, c_n)}{\partial c_k} = 0$$
$$\frac{\partial}{\partial c_k} \left[ \sum_{i=1}^m \left( y_i - \sum_{j=0}^n c_j \phi_j(x_i) \right)^2 \right] = 0$$
$$2 \left[ \sum_{i=1}^m \left( y_i - \sum_{j=0}^n c_j \phi_j(x_i) \right) (-\phi_k(x_i)) \right] = 0$$

# Caso Discreto – Formulação Geral

$$2 \left[ \sum_{i=1}^m \left( y_i - \sum_{j=0}^n c_j \phi_j(x_i) \right) (-\phi_k(x_i)) \right] = 0$$

$$\left[ \sum_{i=1}^m \left( y_i - \sum_{j=0}^n c_j \phi_j(x_i) \right) \phi_k(x_i) \right] = 0$$

$$\sum_{i=1}^m \left[ y_i \phi_k(x_i) - \sum_{j=0}^n c_j \phi_j(x_i) \phi_k(x_i) \right] = 0$$

$$\sum_{i=1}^m y_i \phi_k(x_i) - \sum_{i=1}^m \sum_{j=0}^n c_j \phi_j(x_i) \phi_k(x_i) = 0$$

$$\sum_{i=1}^m \sum_{j=0}^n c_j \phi_j(x_i) \phi_k(x_i) = \sum_{i=1}^m y_i \phi_k(x_i)$$

## Caso Discreto – Formulação Geral

- ▶ Invertendo a ordem dos somatórios, então

$$\sum_{j=0}^n \sum_{i=1}^m c_j \phi_j(x_i) \phi_k(x_i) = \sum_{i=1}^m y_i \phi_k(x_i)$$

$$\sum_{j=0}^n c_j \sum_{i=1}^m \phi_j(x_i) \phi_k(x_i) = \sum_{i=1}^m y_i \phi_k(x_i)$$

- ▶ Utilizando a notação de produto interno:

$$\langle \phi_j; \phi_k \rangle = \langle \phi_k; \phi_j \rangle = \sum_{i=1}^m \phi_j(x_i) \phi_k(x_i)$$

$$\langle y; \phi_k \rangle = \langle \phi_k; y \rangle = \sum_{i=1}^m y_i \phi_k(x_i)$$

## Caso Discreto – Formulação Geral

- ▶ Invertendo a ordem dos somatórios, então

$$\sum_{j=0}^n \sum_{i=1}^m c_j \phi_j(x_i) \phi_k(x_i) = \sum_{i=1}^m y_i \phi_k(x_i)$$

$$\sum_{j=0}^n c_j \sum_{i=1}^m \phi_j(x_i) \phi_k(x_i) = \sum_{i=1}^m y_i \phi_k(x_i)$$

- ▶ Utilizando a notação de produto interno:

$$\langle \phi_{\mathbf{j}}; \phi_{\mathbf{k}} \rangle = \langle \phi_{\mathbf{k}}; \phi_{\mathbf{j}} \rangle = \sum_{i=1}^m \phi_j(x_i) \phi_k(x_i)$$

$$\langle \mathbf{y}; \phi_{\mathbf{k}} \rangle = \langle \phi_{\mathbf{k}}; \mathbf{y} \rangle = \sum_{i=1}^m y_i \phi_k(x_i)$$

# Caso Discreto – Formulação Geral

- Assim, obtém-se

$$\sum_{j=0}^n c_j \langle \phi_{\mathbf{k}}; \phi_{\mathbf{j}} \rangle = \langle \phi_{\mathbf{k}}; \mathbf{y} \rangle, \quad k = 0, \dots, n$$

- Ou seja, os coeficientes  $c_0, \dots, c_n$  do modelo são determinados resolvendo o seguinte sistema de equações lineares  $(n+1) \times (n+1)$

$$\begin{bmatrix} \langle \phi_0; \phi_0 \rangle & \langle \phi_0; \phi_1 \rangle & \dots & \langle \phi_0; \phi_n \rangle \\ \langle \phi_1; \phi_0 \rangle & \langle \phi_1; \phi_1 \rangle & \dots & \langle \phi_1; \phi_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \phi_n; \phi_0 \rangle & \langle \phi_n; \phi_1 \rangle & \dots & \langle \phi_n; \phi_n \rangle \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \langle \phi_0; \mathbf{y} \rangle \\ \langle \phi_1; \mathbf{y} \rangle \\ \vdots \\ \langle \phi_n; \mathbf{y} \rangle \end{bmatrix}$$

# Caso Discreto – Formulação Geral

- Assim, obtém-se

$$\sum_{j=0}^n c_j \langle \phi_k; \phi_j \rangle = \langle \phi_k; \mathbf{y} \rangle, \quad k = 0, \dots, n$$

- Ou seja, os coeficientes  $c_0, \dots, c_n$  do modelo são determinados resolvendo o seguinte sistema de equações lineares  $(n+1) \times (n+1)$

$$\begin{bmatrix} \langle \phi_0; \phi_0 \rangle & \langle \phi_0; \phi_1 \rangle & \dots & \langle \phi_0; \phi_n \rangle \\ \langle \phi_1; \phi_0 \rangle & \langle \phi_1; \phi_1 \rangle & \dots & \langle \phi_1; \phi_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \phi_n; \phi_0 \rangle & \langle \phi_n; \phi_1 \rangle & \dots & \langle \phi_n; \phi_n \rangle \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \langle \phi_0; \mathbf{y} \rangle \\ \langle \phi_1; \mathbf{y} \rangle \\ \vdots \\ \langle \phi_n; \mathbf{y} \rangle \end{bmatrix}$$

## Caso Discreto – Formulação Geral

$$\begin{bmatrix} \langle \phi_0; \phi_0 \rangle & \langle \phi_0; \phi_1 \rangle & \dots & \langle \phi_0; \phi_n \rangle \\ \langle \phi_1; \phi_0 \rangle & \langle \phi_1; \phi_1 \rangle & \dots & \langle \phi_1; \phi_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \phi_n; \phi_0 \rangle & \langle \phi_n; \phi_1 \rangle & \dots & \langle \phi_n; \phi_n \rangle \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \langle \phi_0; \mathbf{y} \rangle \\ \langle \phi_1; \mathbf{y} \rangle \\ \vdots \\ \langle \phi_n; \mathbf{y} \rangle \end{bmatrix}$$

- ▶ O sistema é chamado de sistema de equações normais ou sistema normal
- ▶ Se as funções  $\phi_j(x)$  forem tais que os vetores  $\phi_j$  sejam linearmente independentes então o sistema tem solução única
- ▶ Nota-se que a matriz é simétrica

$$\langle \phi_j; \phi_k \rangle = \langle \phi_k; \phi_j \rangle$$



# Método dos Mínimos Quadrados

- ▶ Seja a matriz  $\mathbf{A}$  definida como

$$\mathbf{A} = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_n(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \dots & \phi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_m) & \phi_1(x_m) & \dots & \phi_n(x_m) \end{bmatrix}$$

- ▶ E o vetor  $\mathbf{y}$  como

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

# Método dos Mínimos Quadrados

- ▶ Então o sistema de equações normais do Método dos Mínimos Quadrados fica como

$$\mathbf{A}^T \mathbf{A} \mathbf{c} = \mathbf{A}^T \mathbf{y}$$

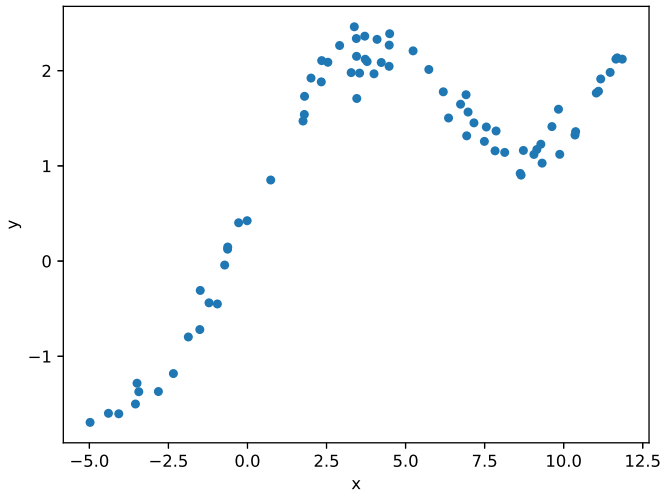
- ▶ Portanto, os coeficientes do modelo podem ser obtidos apenas resolvendo um sistema linear

# Método dos Mínimos Quadrados: Python

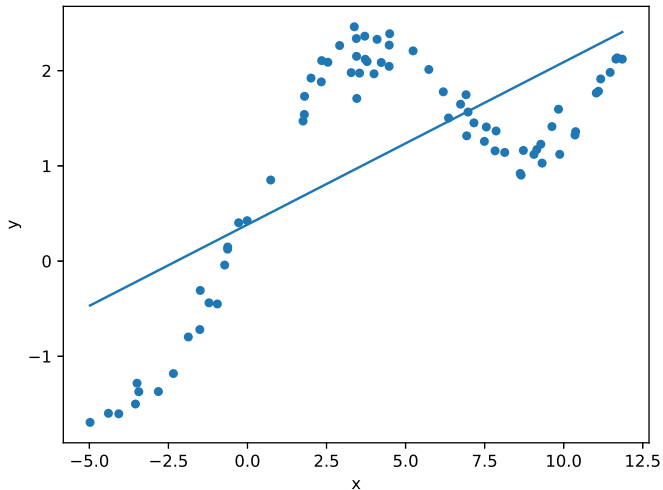
- ▶ Arquivo dos dados de entrada: `reg_2_tr_X.dat`
- ▶ Arquivo dos valores esperados: `reg_2_tr_Y.dat`
- ▶ <http://pandas.pydata.org/pandas-docs/stable/visualization.html>

x	y
4.4970249	2.3881414
1.7572682	1.4709548
2.0093448	1.9221703
-1.5073673	-0.71916443
5.2374846	2.2081153
...	...
-1.4937856	-0.30838877
2.3348046	1.8823554
4.4828742	2.268271
8.7256175	1.1620196

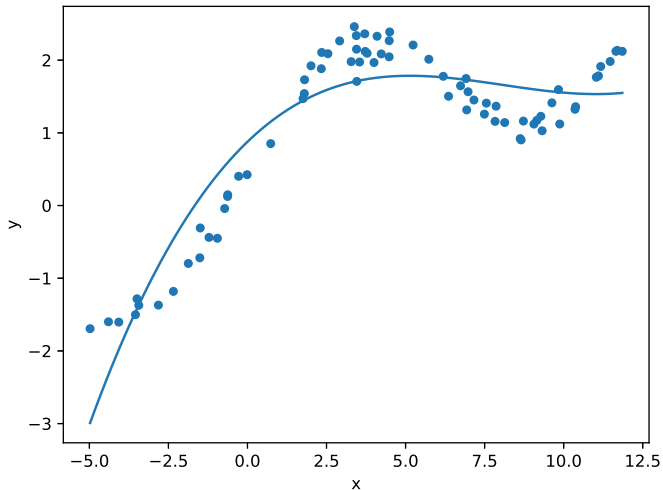
# Método dos Mínimos Quadrados: Python



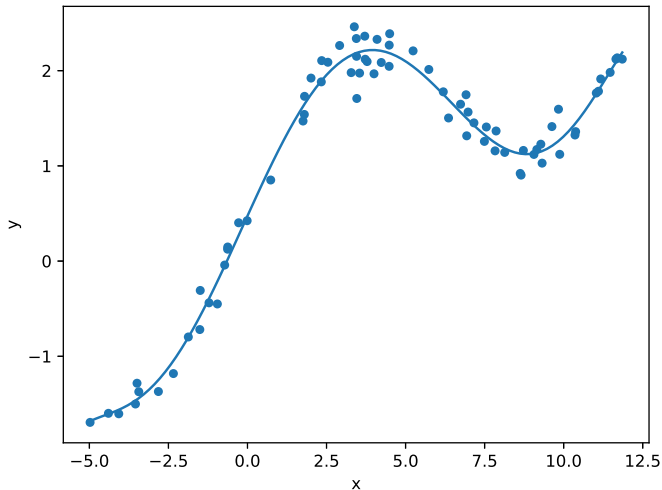
# Método dos Mínimos Quadrados: Reta



# Método dos Mínimos Quadrados: Polinômio de 3o grau



## Método dos Mínimos Quadrados: Polinômio de 6o grau



# Regressão Ridge



# Regressão Ridge

- ▶ Modelo que impõem uma penalidade no Método dos Mínimos Quadrados

$$\min_{\mathbf{c}} ||\mathbf{Ac} - \mathbf{y}||_2^2 + \alpha ||\mathbf{c}||_2^2$$

- ▶  $\alpha$  controla a penalização pelo aumento do tamanho do vetor de coeficientes
- ▶ Quanto maior o  $\alpha$ , mais robusto à colinearidade o método é
- ▶ Assim como o Método dos Mínimos Quadrados, o Ridge pode ser usado para classificação
  - ▶ Caso não abordado aqui
- ▶ A implementação do Scikit Learn é para um modelo linear

# Regressão Lasso

# Regressão Lasso

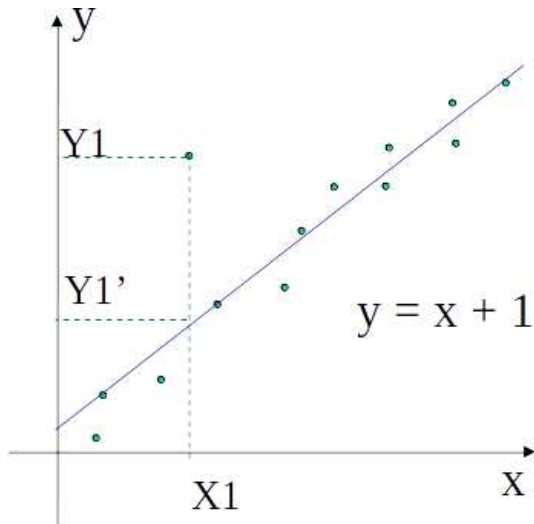
- ▶ Modelo que impõem uma penalidade no Método dos Mínimos Quadrados

$$\min_{\mathbf{c}} \frac{1}{2n} \|\mathbf{A}\mathbf{c} - \mathbf{y}\|_2^2 + \alpha \|\mathbf{c}\|_1$$

- ▶  $n$  é o número de instâncias da base de dados
- ▶  $\alpha$  controla a penalização pelo aumento do tamanho do vetor de coeficientes
- ▶ Útil em alguns contextos dada a tendência em preferir soluções com poucos coeficientes não nulos, reduzindo o número de variáveis necessárias
- ▶ Também pode ser usado para classificação
  - ▶ Caso não abordado aqui
- ▶ A implementação do Scikit Learn é para um modelo linear e usa o Método das Coordenadas Descendentes para resolver a otimização

# Avaliação de Modelos

# Avaliação de Modelos



# Avaliação de Modelos

- ▶ Sejam  $y_i$  e  $\hat{y}_i$  os  $i$ -ésimos valores esperados e preditos
- ▶ Erro Absoluto Médio (Mean Absolute Error, MAE)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- ▶ Erro Absoluto Médio Percentual (Mean Absolute Percentage Error, MAPE)

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{|y_i|}$$

# Avaliação de Modelos

- ▶ Erro Quadrático Médio (Mean Square Error, MSE)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- ▶ Raiz Quadrada do Erro Quadrático Médio (Root Mean Square Error, RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- ▶ Raiz Quadrada do Erro Quadrático Médio em Logaritmo (Root Mean Squared Log Error, RMSLE)

$$\text{RMSLE} = \sqrt{\frac{1}{n} \sum_{i=1}^n [\log(\hat{y}_i + 1) - \log(y_i + 1)]^2}$$

# Avaliação de Modelos

- ▶ Seja  $\bar{y}$  a média dos valores de  $y$
- ▶ Coeficiente de determinação ( $R^2$ )

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

- ▶ Outras métricas e exemplos disponíveis em:  
[https://scikit-learn.org/stable/modules/model\\_evaluation.h](https://scikit-learn.org/stable/modules/model_evaluation.html)



# Outros Métodos e Contextos

# Outros Métodos e Contextos

- ▶ Existem outros métodos para resolver problemas de regressão
  - ▶ Redes Neurais Artificiais
  - ▶ Regressão por Vetores de Suporte (SVR)
  - ▶ Árvores de Decisão para Regressão
  - ▶ Modelos Lineares Generalizados (GLM)
  - ▶ :
  
- ▶ Aplicações em contexto temporal: Séries Temporais
  - ▶ Médias Móveis
  - ▶ ARMA
  - ▶ ARIMA
  - ▶ SARIMA