INF623

2024/1

UFV

Inteligência Artificial

A24: Aprendizado supervisionado

Plano de aula

- Aprendizado supervisionado
- Espaço de classes e características
- Classificação vs. Regressão
- Espaço de hipóteses
- Funções de perda
- Generalização: subajuste vs. sobreajuste

UFV

Exemplo 1: detecção de spam

Considere o problema de identificar se um determinado email é spam ou não. Como você escreveria um algoritmo para resolver esse problema?

Email 1

Olá Lucas,

A reunião do colegiado do DPI será na próxima segunda-feira, às 8:00h.

Atenciosamente

?

→ **Não spam**

Email 2

Olá Lucas,

Condição especial no plano de saúde, preços até 40% menores!

APROVEITE!

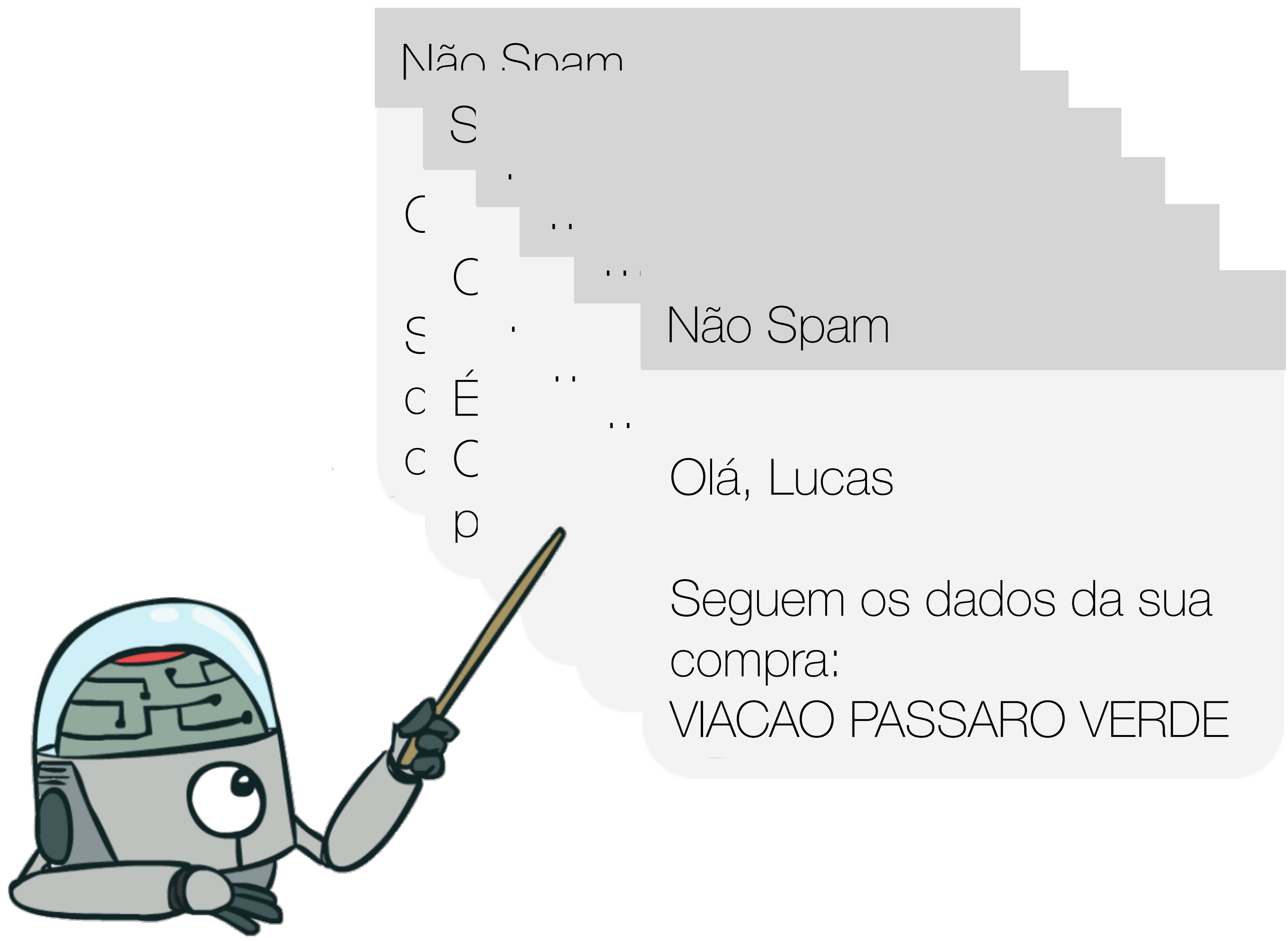
?

→ **Spam**

UFV

Agentes racionais de aprendizado supervisionado

Para resolver problemas desse tipo, um agente assume que o mundo é representado por uma distribuição desconhecida $P(X, Y)$, e que o **objetivo é encontrar uma função** h a partir de um **conjunto de dados** D, tal que, para um novo exemplo $(x', y') \notin D$ amostrado de P, temos $h(x) \approx y'$



- O conjunto de dados D é composto por exemplos (x, y), onde x é um **vetor de características** e y é um **rótulo** (ou classe)
- A **função** $\hat{y} = h(x)$ mapeia vetores de entrada x em rótulos $\hat{y}$ (previsão)
- Queremos encontrar a função $h(x)$ com menor **erro** de previsão em exemplos novos $(x', y') \notin D$

Formalização de aprendizado supervisionado

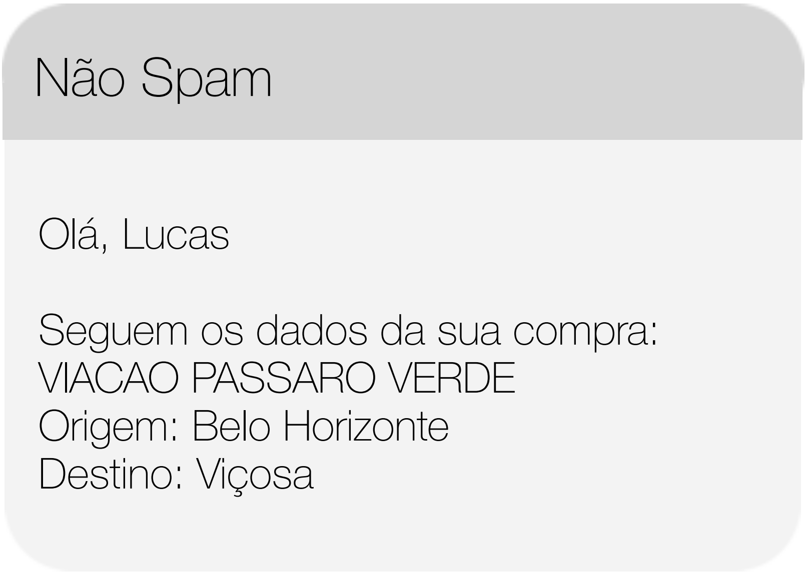
Um problema de **aprendizado supervisionado** pode ser formalmente definido por:

- Um conjunto de dados $D = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq \mathbb{R}^d \times C$, onde:
  - x_i é o vetor de características do $i-$ésimo exemplo
  - y_i é o rótulo (ou classe) do i-ésimo exemplo
  - $\mathbb{R}^d$ é o espaço de características
  - C é o espaço de classes

UFV

Exemplos de espaços de classes

$$D = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq \mathbb{R}^d \times C$$



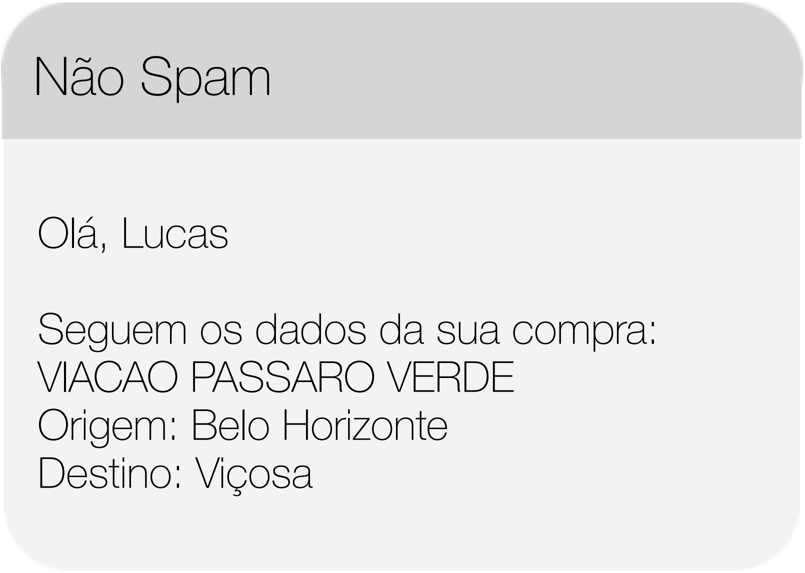
Detecção de Spam
Classificação binária
C = {0,1}

Reconhecimento de Dígitos Manuscritos
Classificação multi-classe
C = {0,1,2,3,4,5,6,7,8,9}

Previsão de Preços de Imóveis
Regressão
C = Conjunto dos Reais (R)

Exemplos de vetores de características

$$D = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq \mathbb{R}^d \times C$$



Texto (*não estruturado*)

x_i : número de ocorrências da i-ésima palavra de um dicionário

d ~ 100.000 - 10M

Imagem (*não estruturado*)

x_i : valor do i-ésimo pixel da imagem achatada

d ~ 100.000 - 10M

Dados Tabulares (Estruturados)

x_i : valor da i-ésima coluna de uma tabela

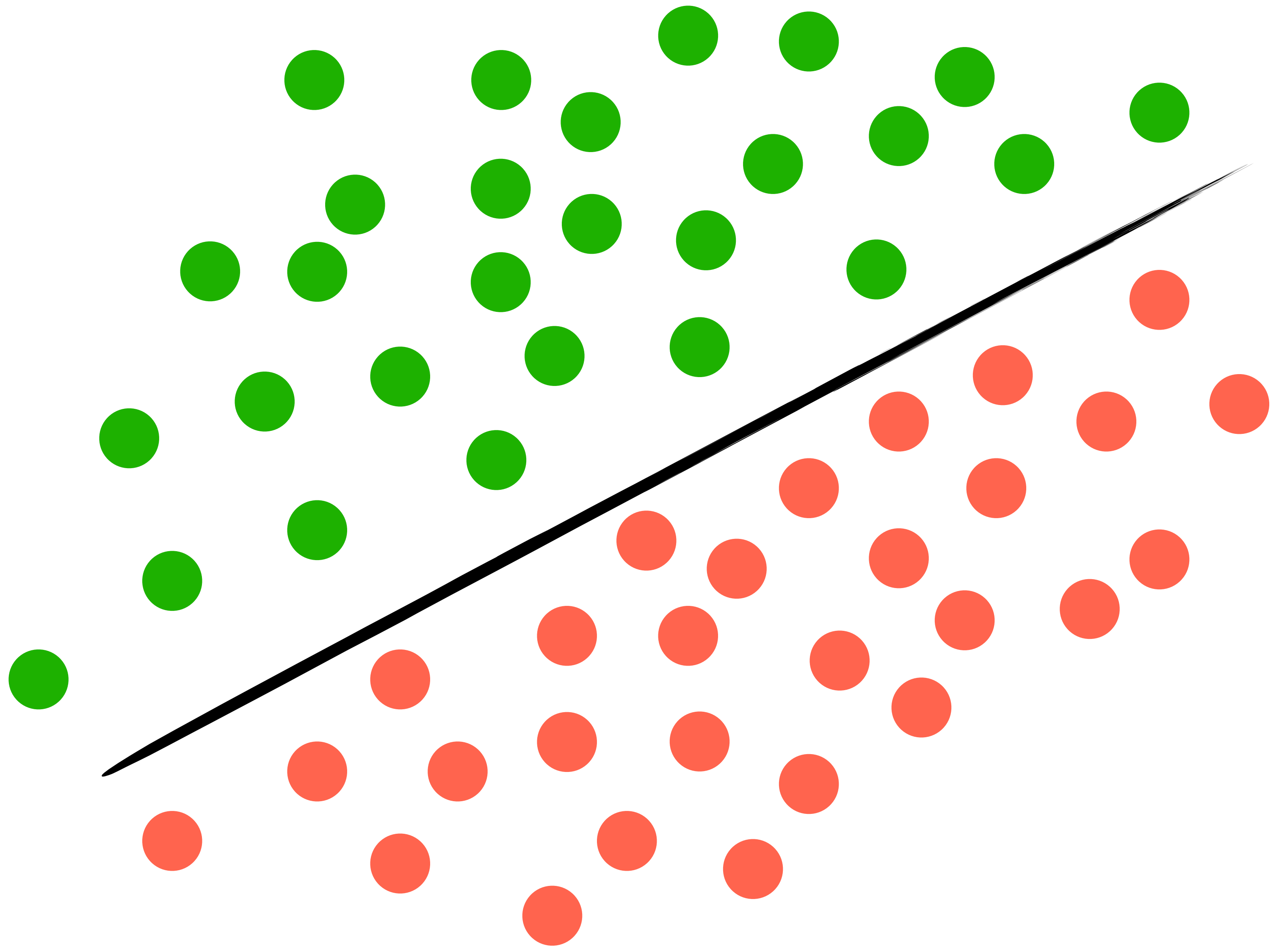
x_1: tamanho, x_2: localização, ..., x_n: número de quartos

d igual ao número de colunas

Classificação vs Regressão

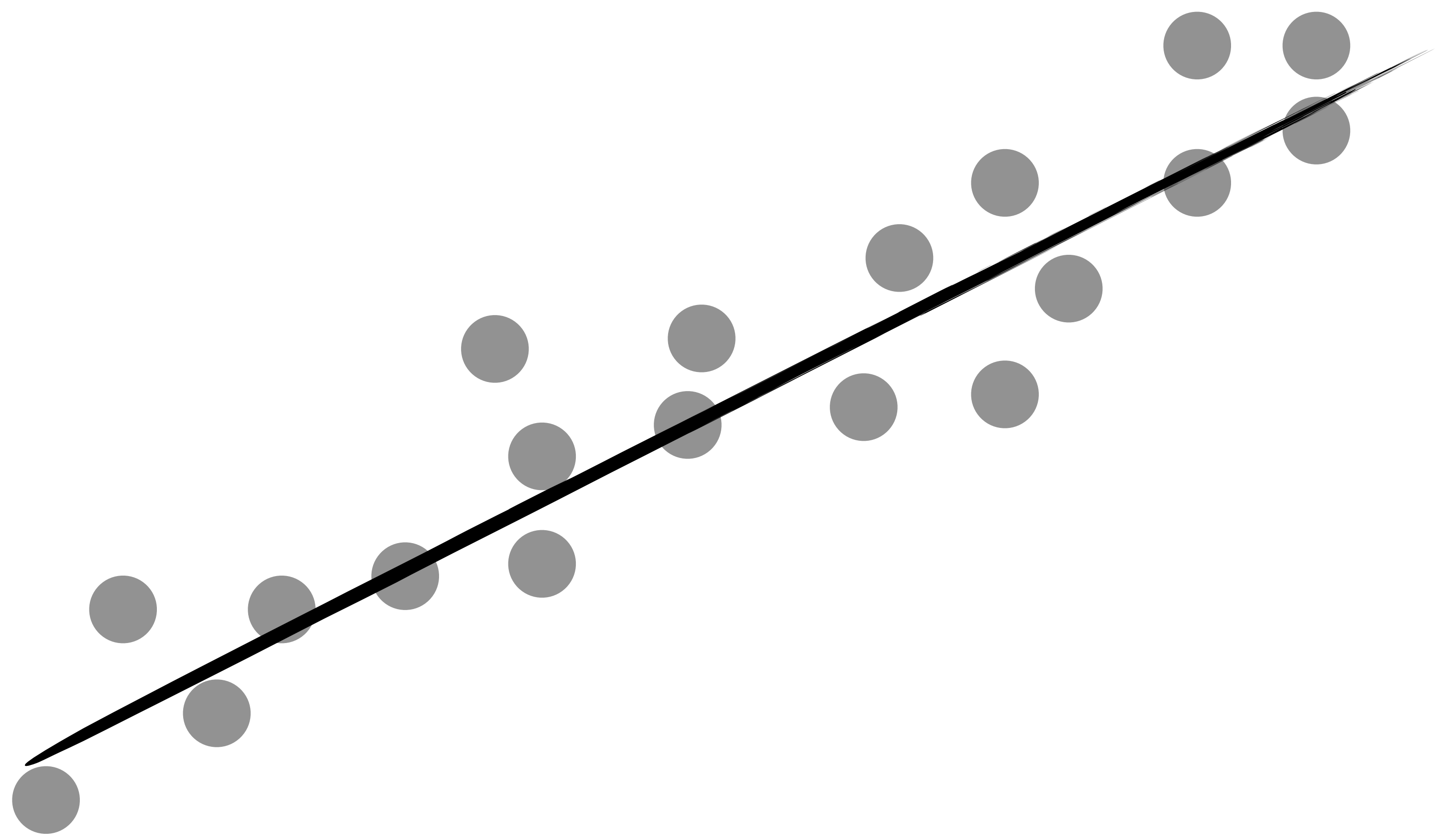
$$D = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq \mathbb{R}^d \times C$$

Classificação



Encontrar uma função (e.g., linear) que *separa* as classes da melhor forma.

Regressão

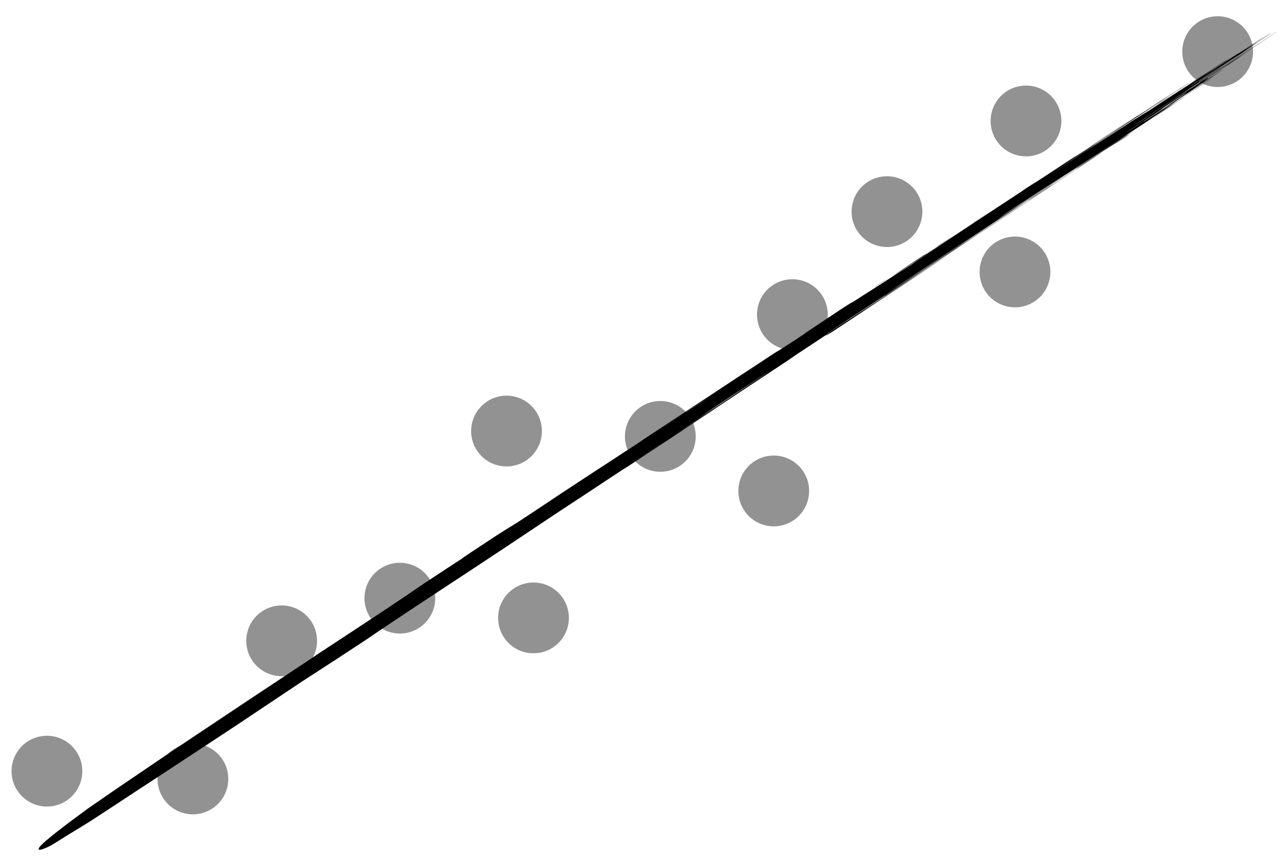


Encontrar uma função (e.g., linear) que se ajusta melhor aos dados

Espaço de hipóteses

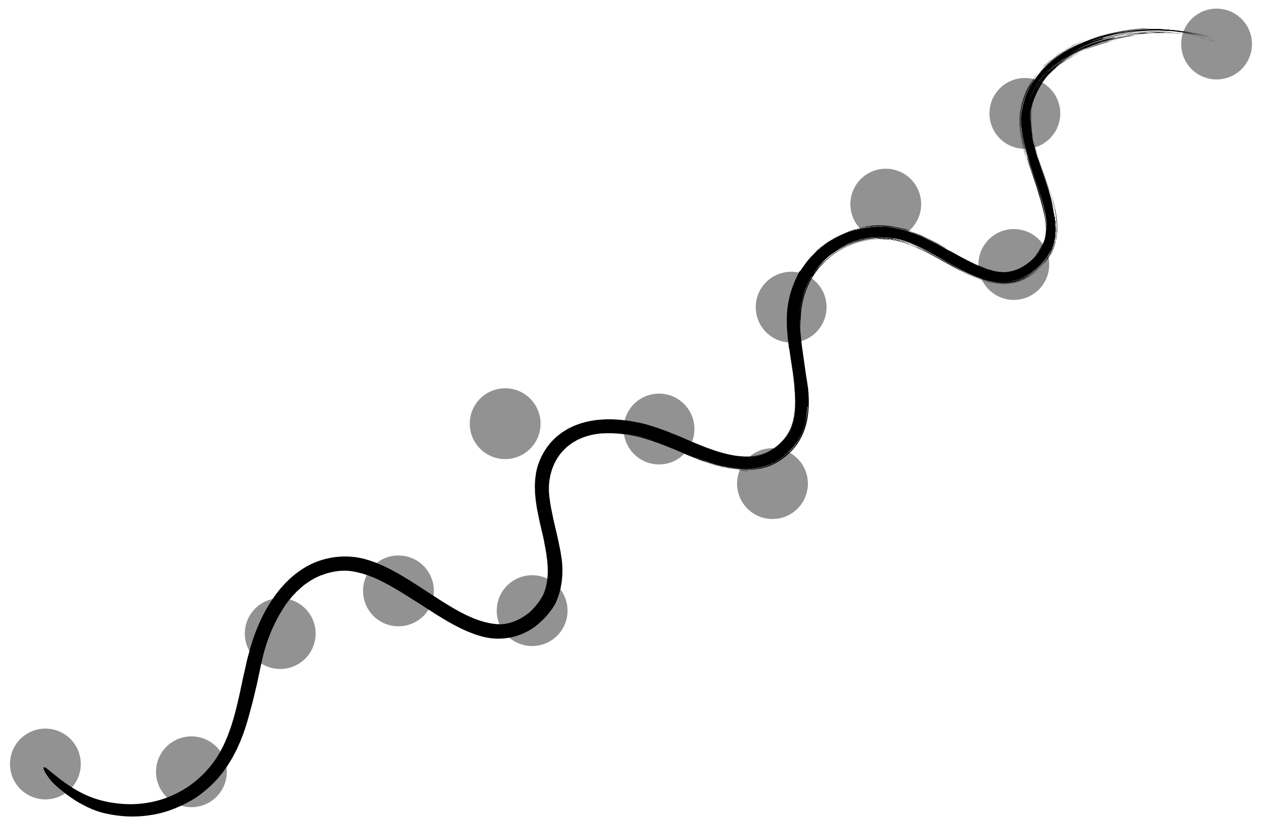
O *espaço de hipóteses* H define o conjunto de funções que um algoritmo de aprendizado supervisionado pode encontrar.

Exemplos:



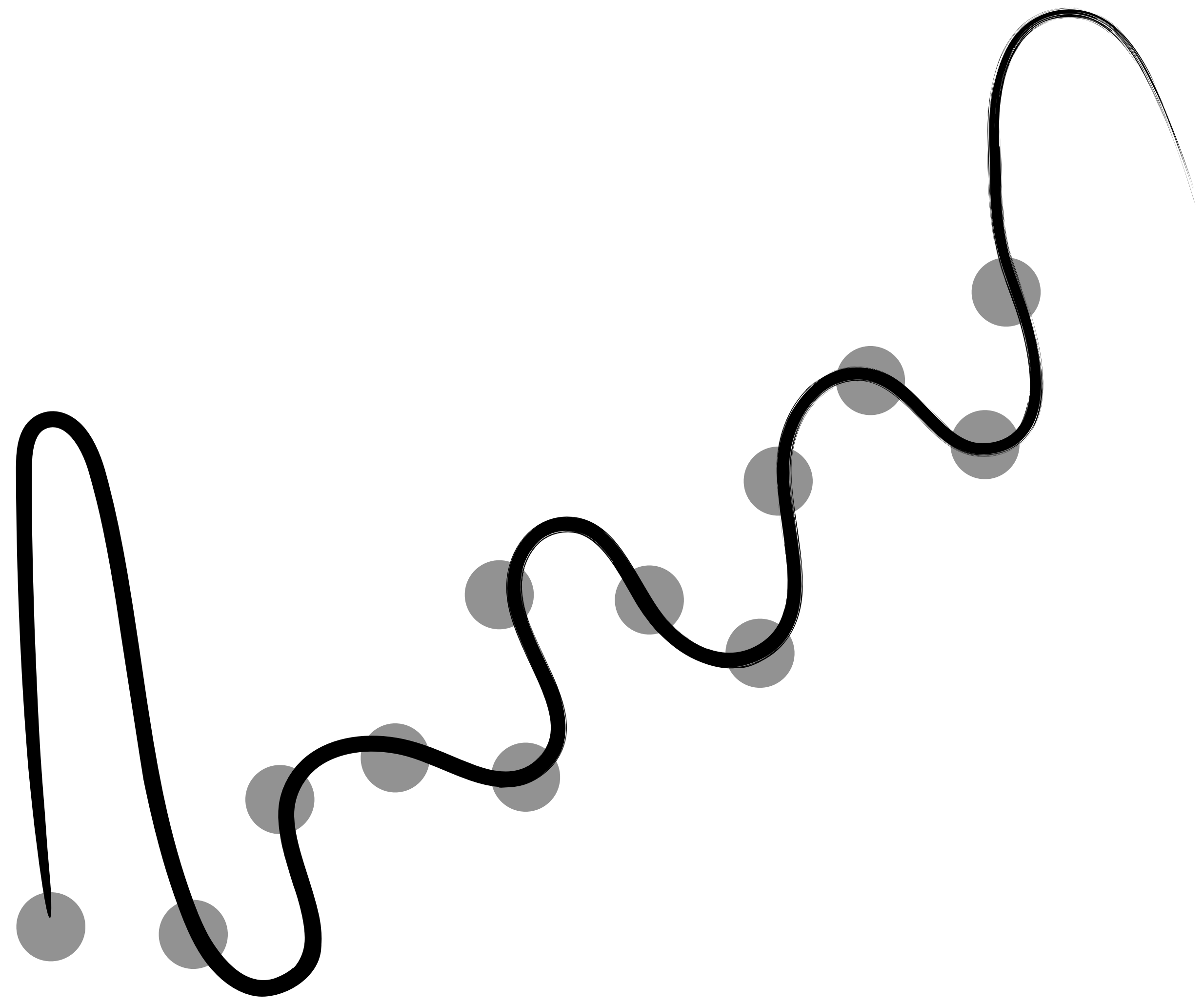
Reta

$$h(x) = w_1 x + w_0$$



Senoide

$$h(x) = w_1 x + sin(w_0 x)$$

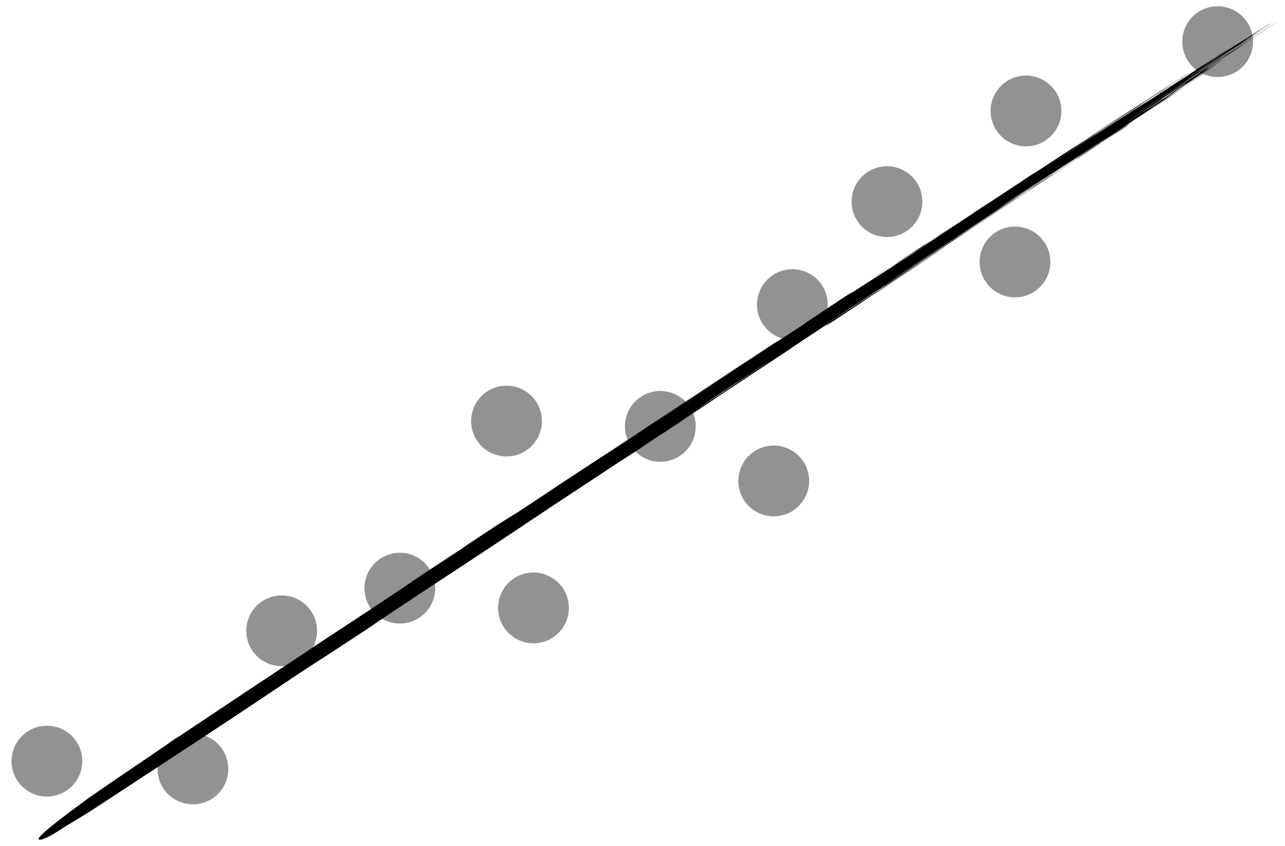


Polinômio de grau 12

$$h(x) = \sum_{i=0}^{12} w_i x^i$$

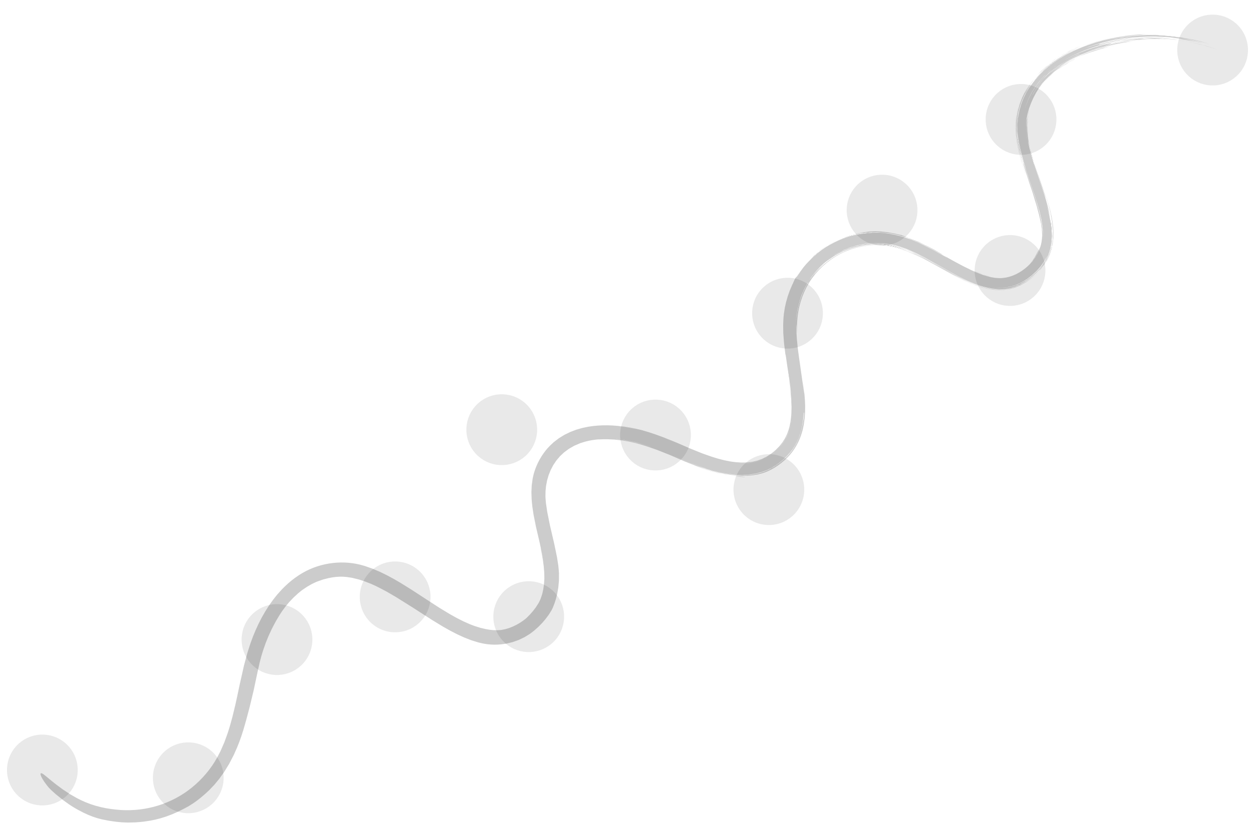
Espaço de hipóteses

Assumindo, por exemplo, uma reta como hipótese, precisamos ajustar os parâmetros w_1 e w_0 para minimizar o erro no conjunto de dados D.



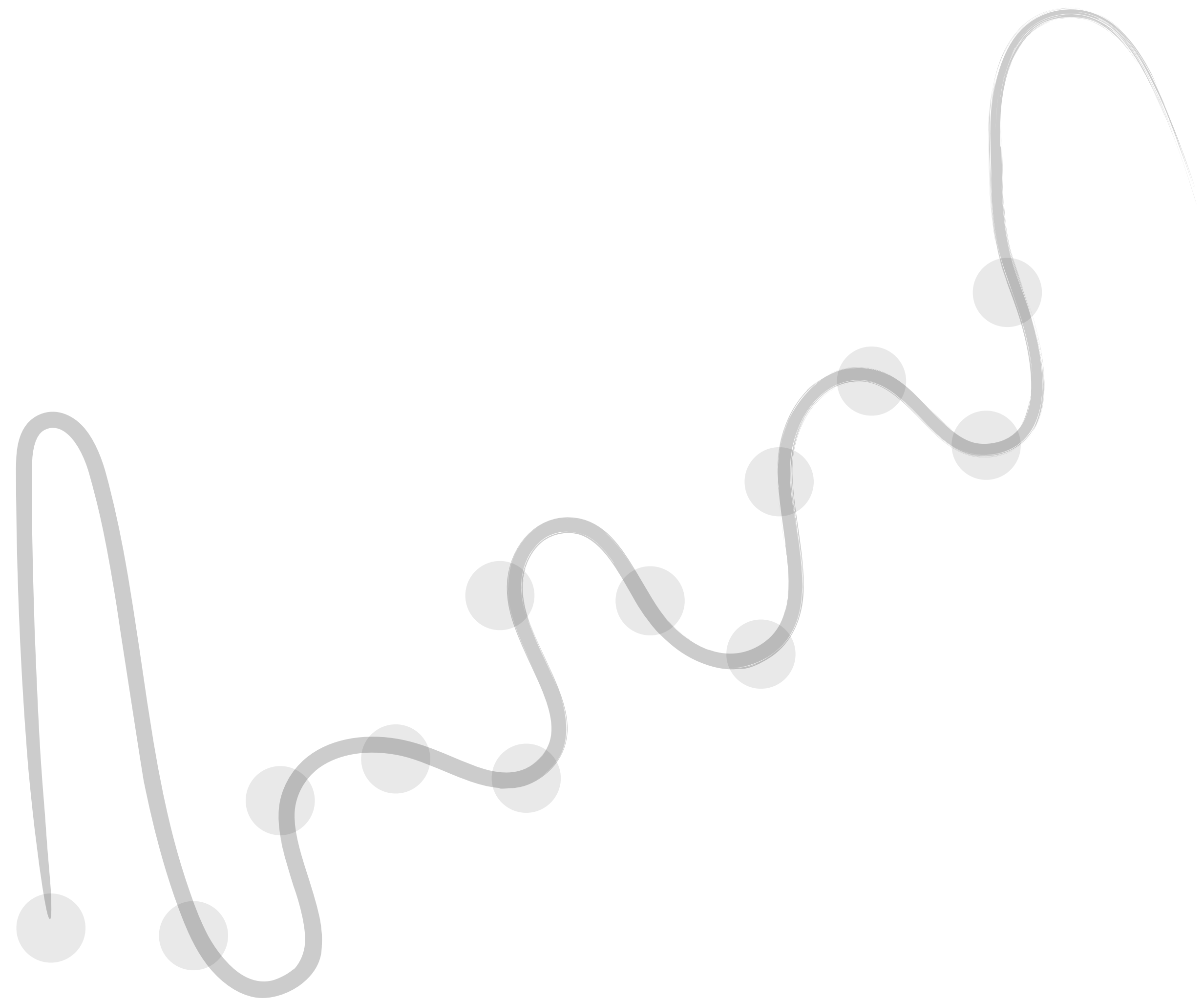
Reta

$$h(x) = w_1 x + w_0$$



Senoide

$$h(x) = w_1 x + sin(w_0 x)$$



Polinômio de grau 12

$$h(x) = \sum_{i=0}^{12} w_i x^i$$

Função de perda

A **função da perda** L avalia uma hipótese $h \in H$ com o conjunto de dados $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$:

- Mede o quão distantes as previsões de $h(x_i)$ estão dos rótulos y_i dos exemplos (x_i, y_i) em D;
- Os valores de perda $L(h)$ são sempre positivos;
- Quanto menor a perda $L(h)$, melhor a hipótese h;
- Uma hipótese com perda $L(h) = 0$ (zero) acerta o rótulo de todos os exemplos em D;
- Tipicamente, a função de perda L é normalizada para que o seu valor seja independente do tamanho m do conjunto de dados.

Exemplos:

- Perda Zero-um
- Perda Quadrática
- Perta Absoluta

Exemplo de função de perda: zero-um

O número de erros que uma hipótese h comete nos exemplos de D.

$$L(h) = \frac{1}{m} \sum_{i=1}^{n} \delta_{h(x_i) \neq y_i} \text{ onde } \delta_{h(x_i) \neq y_i} = \begin{cases} 1, & \text{se } h(x_i) \neq y_i \\ 0, & \text{caso contrário} \end{cases}$$

- Geralmente utilizada para avaliar hipóteses em problemas de classificação
- Não é utilizada para treinar uma hipótese, pois não é diferenciável

Exemplo de função de perda: quadrática

A soma do erro quadrático $(h(x_i) - y_i)^2$ da hipótese h nos exemplos de D.

$$L(h) = \frac{1}{m} \sum_{i=1}^{m} (h(x_i) - y_i)^2$$

- Geralmente utilizada para treinar uma hipótese h em problemas de regressão
- Elevar o erro ao quadrado faz com que exemplos com erros mais altos tenham maior influência no ajuste dos pesos de h

UFV

Exemplo de função de perda: absoluta

A soma do erro absoluto $|h(x_i) - y_i|$ da hipótese h nos exemplos de D.

$$L(h) = \frac{1}{m} \sum_{i=1}^{m} |h(x_i) - y_i|$$

- Geralmente utilizada para treinar uma hipótese h em problemas de regressão
- Exemplos têm influência uniforme no ajuste dos pesos
- Adequada para lidar com ruído nos dados (*outliers*)

Generalização

Dado um espaço de hipóteses H e uma função de perda L, queremos encontrar a hipótese $h \in H$:

$$h = argmin_{h \in H} L(h)$$

Se encontrarmos uma hipótese $h \in H$ com baixa perda em D, como saber se ela também terá baixa perda em novos exemplos $(x', y') \notin D$?

Generalização

Considere a seguinte função "memorizadora":

$$h(x) = \begin{cases} y_i, & \text{se } \exists (x_i, y_i) \in D, \text{tal que}, x = x_i \\ 0, & \text{caso contrário} \end{cases}$$

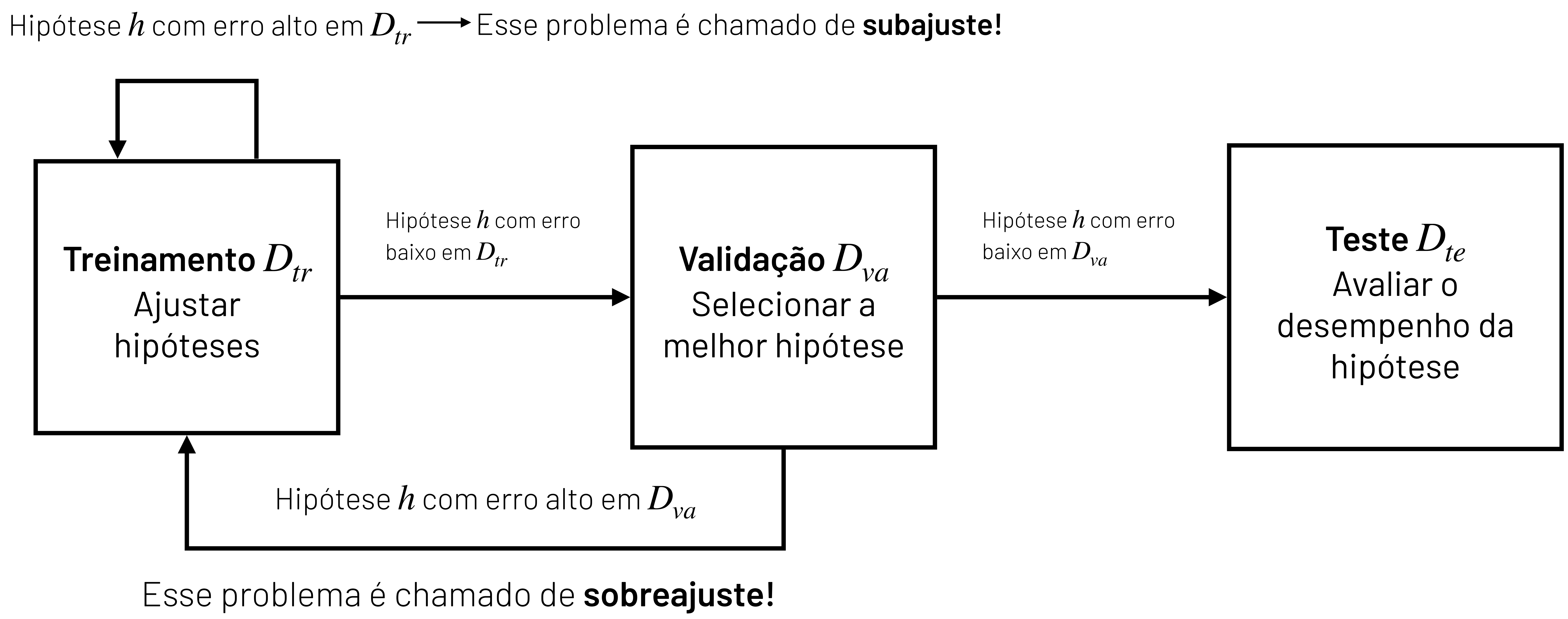
- Perda 0 nos exemplos de D;
- Perda muito alta em exemplos novos!

Esse problema é chamado de **sobreajuste** (*overfit*)!

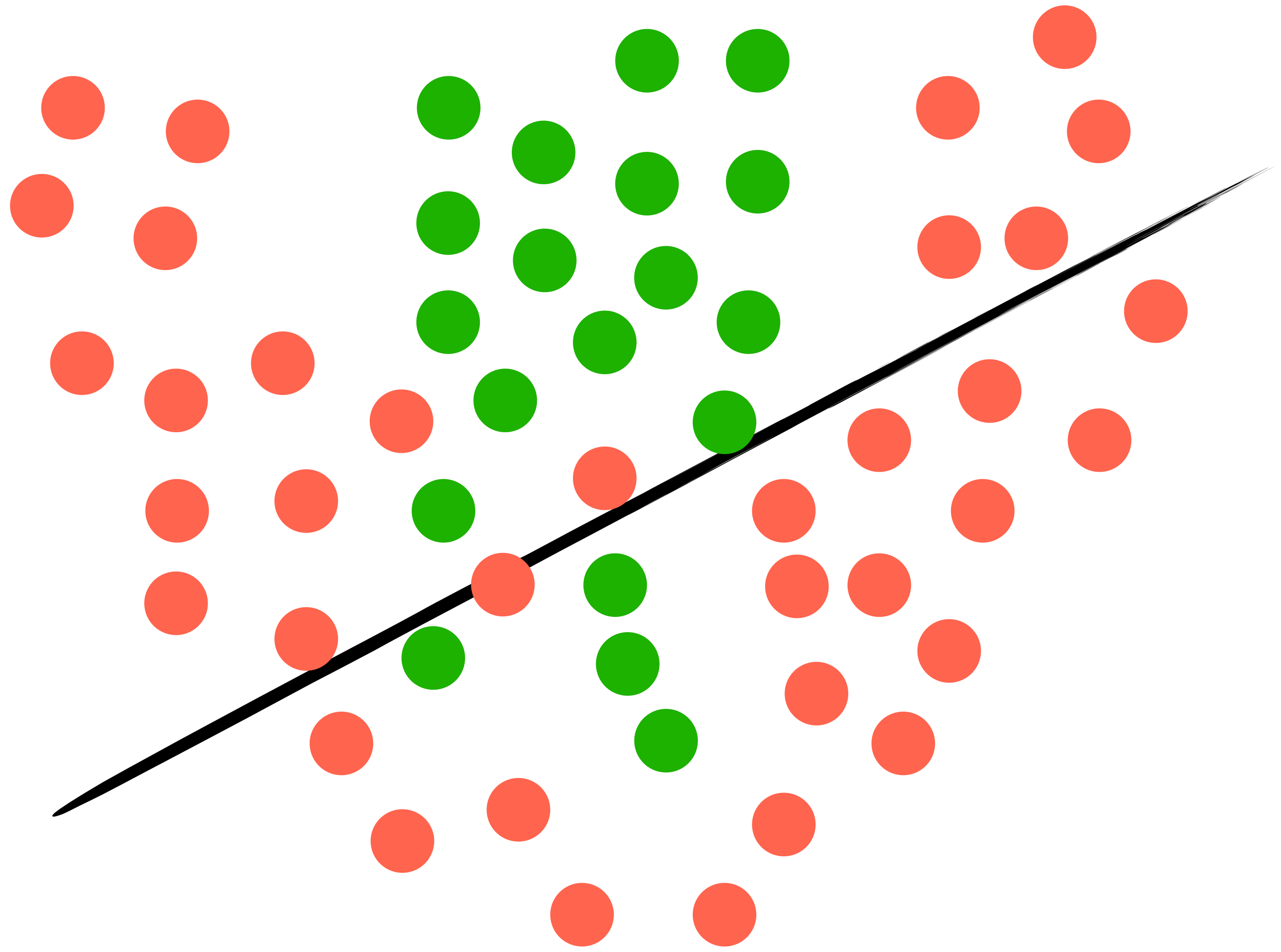
UFV

Subajuste e sobreajuste

Para resolver o problema de sobreajuste, dividimos o conjunto de dados D em três (3) subconjuntos disjuntos D_{tr}, D_{va} e D_{te}:

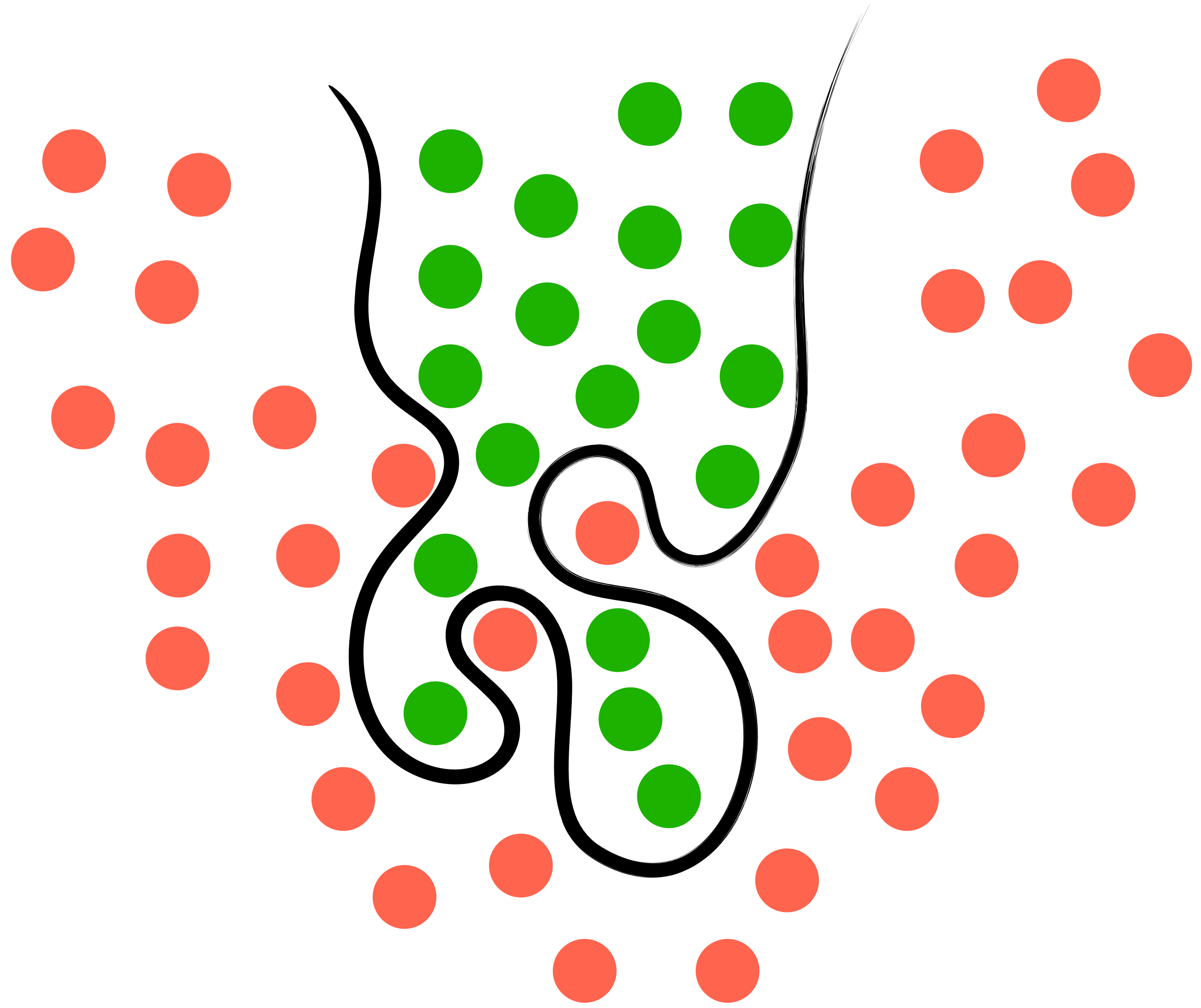


Subajuste (classificação)



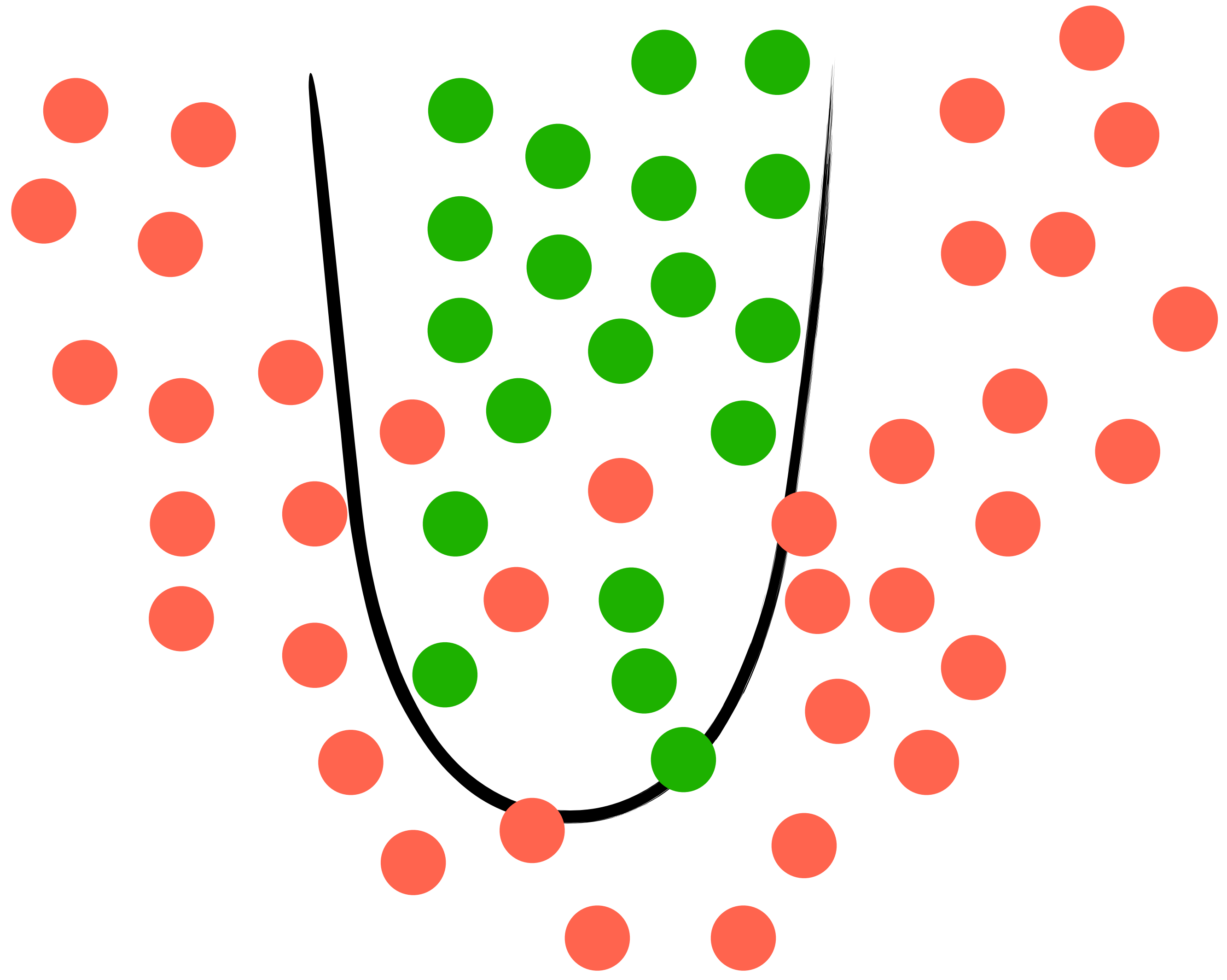
Quando a hipótese se ajusta pouco aos dados de treinamento, aprensentando baixo desempenho de previsão tanto no conjunto de treinamento quanto no de teste.

Sobreajuste (classificação)



Quando a hipótese se ajusta muito aos dados de treinamento, aprensentando alto desempenho de previsão no conjunto de treinamento, mas baixo no conjunto de teste.

Ajuste adequado (classificação)



Quando a hipótese se ajusta bem aos dados de treinamento, aprensentando alto desempenho de previsão tanto no conjunto de treinamento quanto no de teste.

UFV

Generalização

Em aprendizado supervisionado, assumimos três condições sobre o conjunto de dados D:

1. Os exemplos são amostrados de forma **independente e identicamente distribuída (i.i.d)** de $P(X, Y)$;

2. A distribuição $P(X, Y)$ é **estacionária**: não muda ao longo do tempo;

3. Sempre amostramos da **mesma distribuição** $P(X, Y)$, tanto no conjunto de treinamento, quandos nos de validação e teste.

Algoritmos de aprendizado supervisionado

Para encontrar uma função h, um **algoritmo de aprendizado supervisionado** precisa assumir um pressuposto (hipótese) sobre os dados para definir um espaço de funções H restrito que possibilite a busca.

Algorítmos de aprendizado supervisionado

- k-Nearest Neighbors (KNN)
- Naive bayes
- Árvores de decisão
- Suport vector machines (SVMs)
- Regressão linear
- Regressão logística
- Redes neurais

Próxima aula

A25: Aprendizado supervisionado 2

Naive bayes, K-nearest neighbors (kNN) e avaliações de modelos