Aprendizagem Automática em Sistemas Empresariais

PEDRO PEREIRA

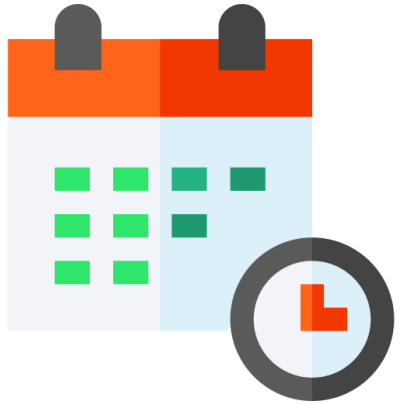
AULA 6

Agenda

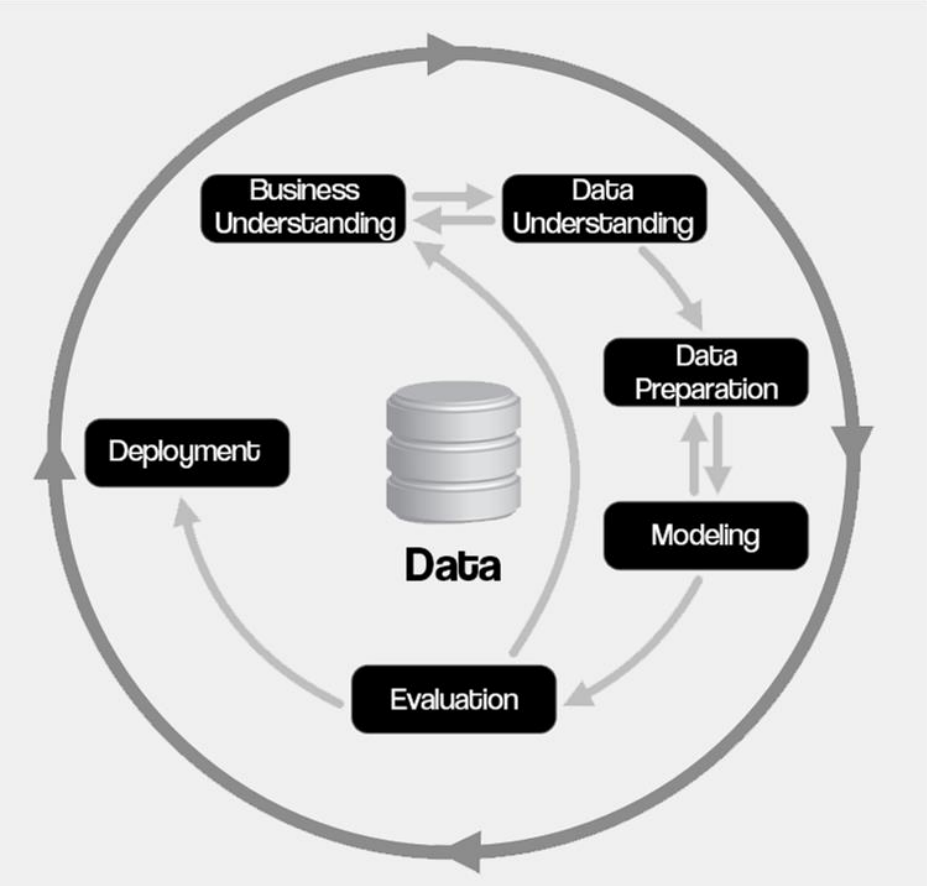
CRISP-DM: Modelação e Avaliação – Parte 2

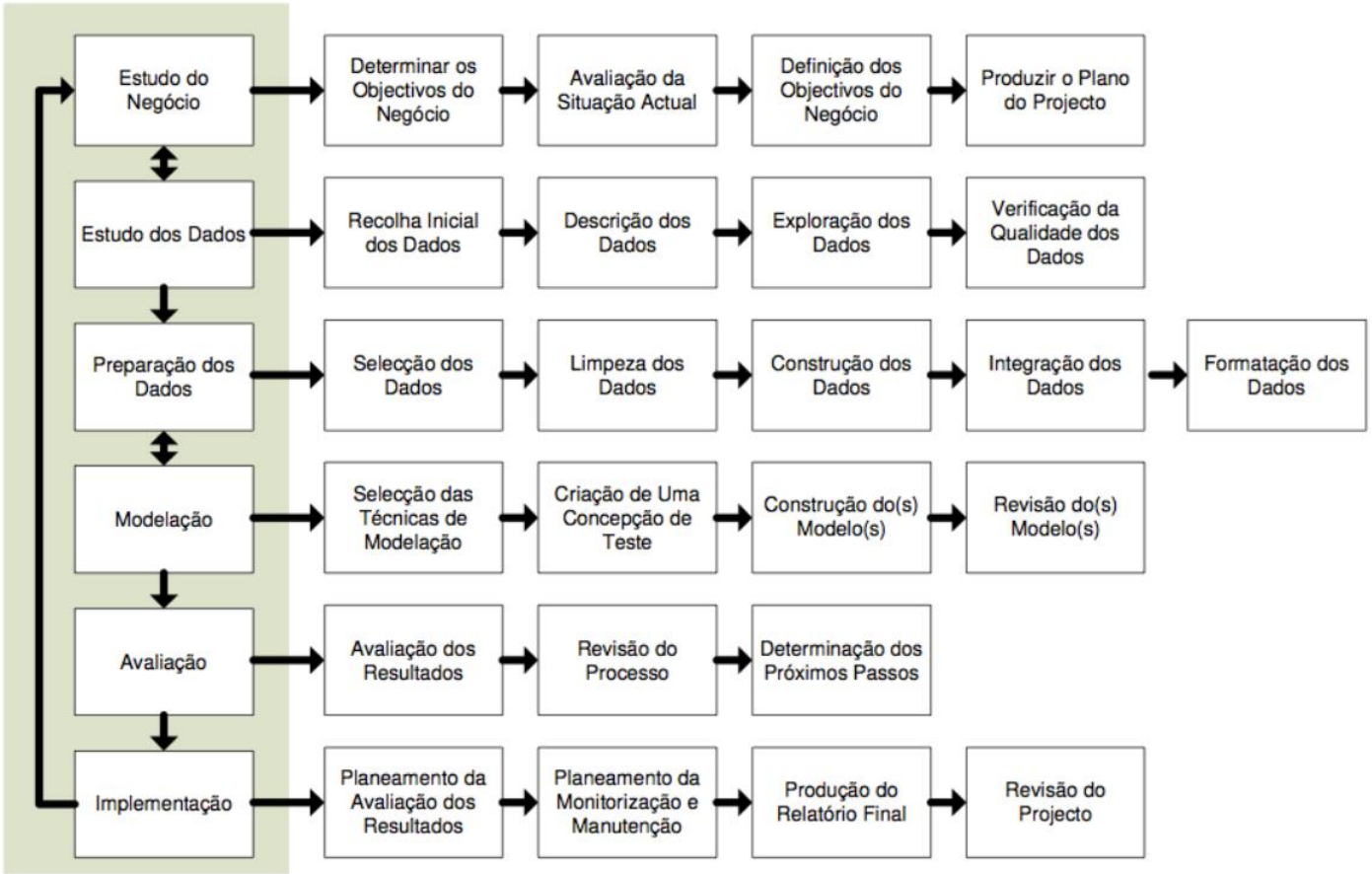
- Regressão
- Algoritmos de ML para regressão
- Métricas de regressão
- Demonstração

Acompanhamento ao projeto

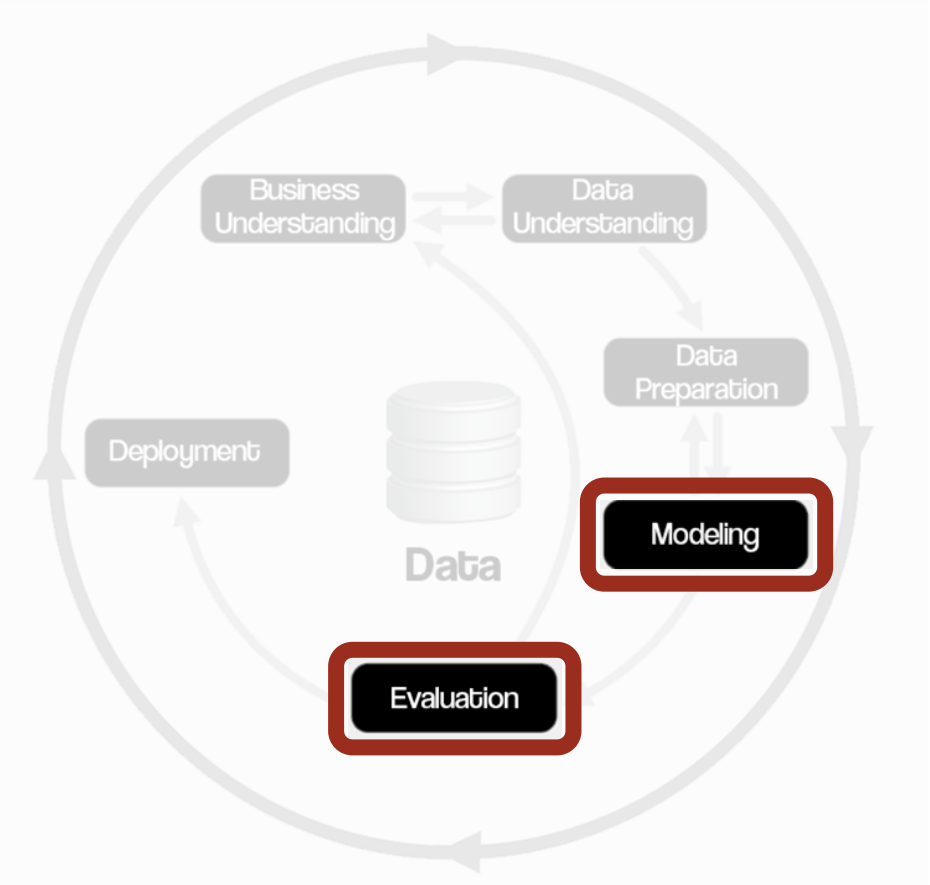


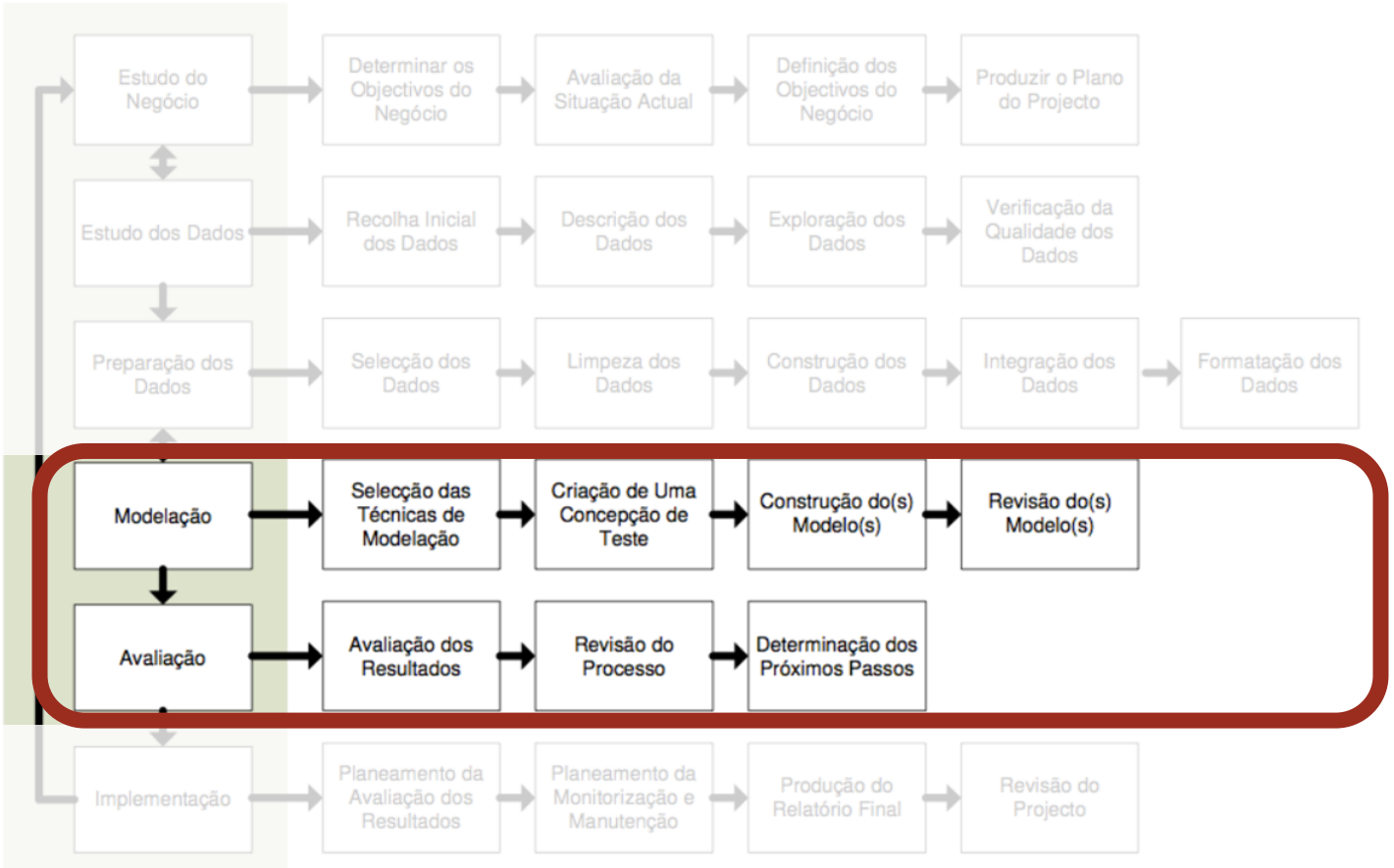
Cross Industry Process for Data Mining (CRISP-DM)



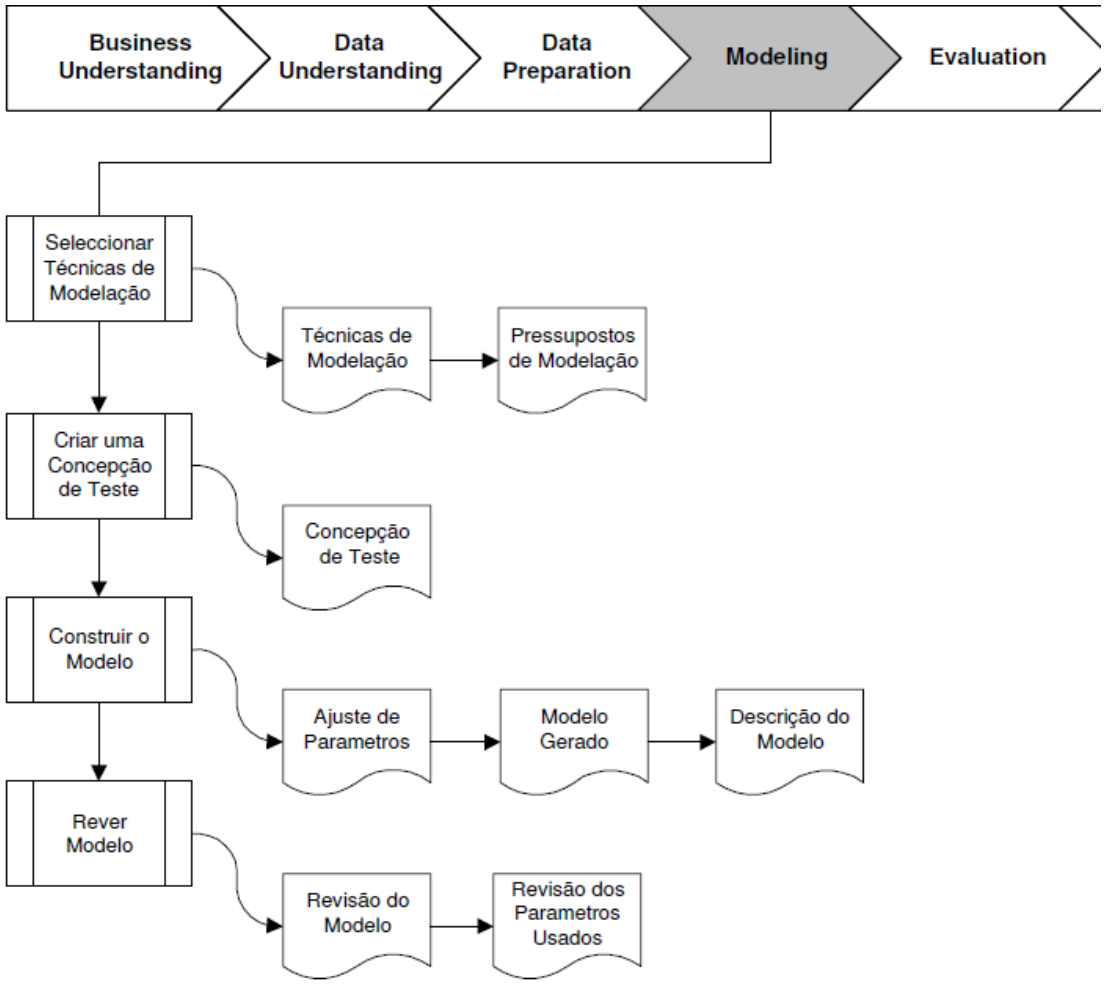


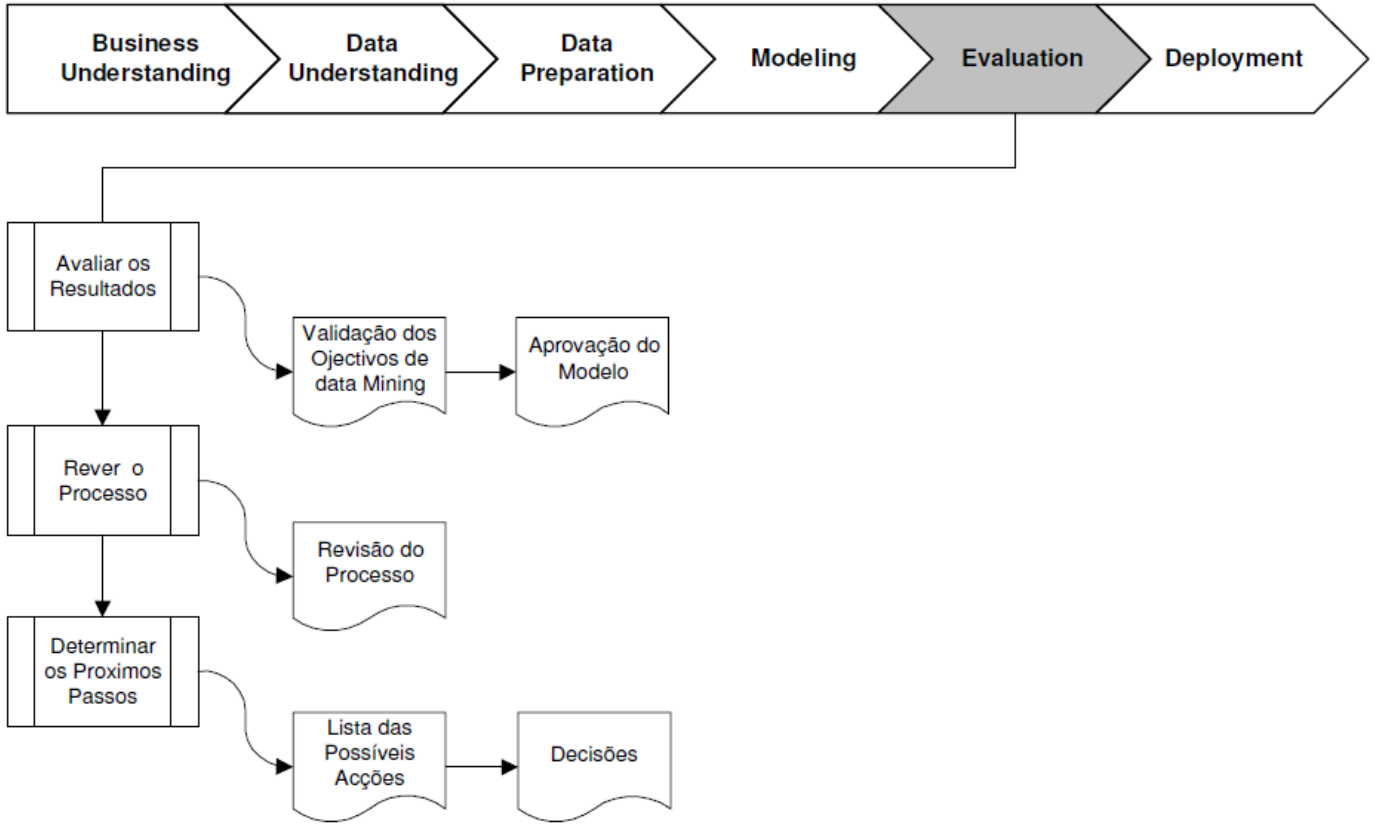
CRISP-DM – Modelação e Avaliação – Parte 2





CRISP-DM – Atividades da Modelação e Avaliação





CRISP-DM – Modelação (regressão)

Regressão – prever o valor de uma variável numérica a partir de diferente variáveis independentes.

Diferentes variantes na regressão:

- **Regressão pura** – prever um valor (*output*) com base num conjunto de variáveis (ex.: prever o preço de um carro usado com base nas suas características).
- **Previsão de Séries Temporais** (valores ordenados no tempo) – prever um valor com base nos seus valores anteriores (ex.: entradas de clientes numa loja, tráfego de internet, número de visualizações de um vídeo).
- **Regressão Multi-target** (dois ou mais *outputs*) – prever simultaneamente mais do que um valor (ex.: prever a composição de um produto).
- **Regressão Ordinal (A<B<C<D) –** prever uma classe quando o valor é ordinal (ex.: prever a satisfação de um cliente {muito insatisfeito < insatisfeito < satisfeito < muito satisfeito}).

Modelação – Algoritmos de regressão

K-Nearest Neighbors (KNN);

Regressão Linear;

ARIMA (séries temporais);

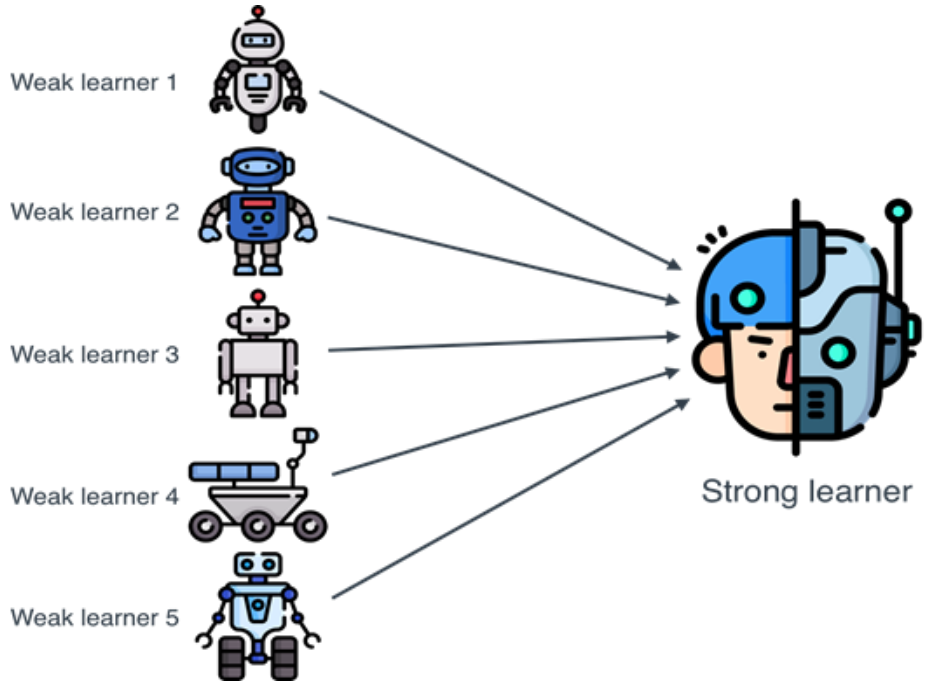
Árvores de Decisão (DT);

Máquinas de Vetor de Suporte (SVM);

Redes Neuronais Artificiais (ANN);

Ensembles (XGBoost, AdaBoost, *Random Forest*, …);

Automated Machine Learning (AutoML).

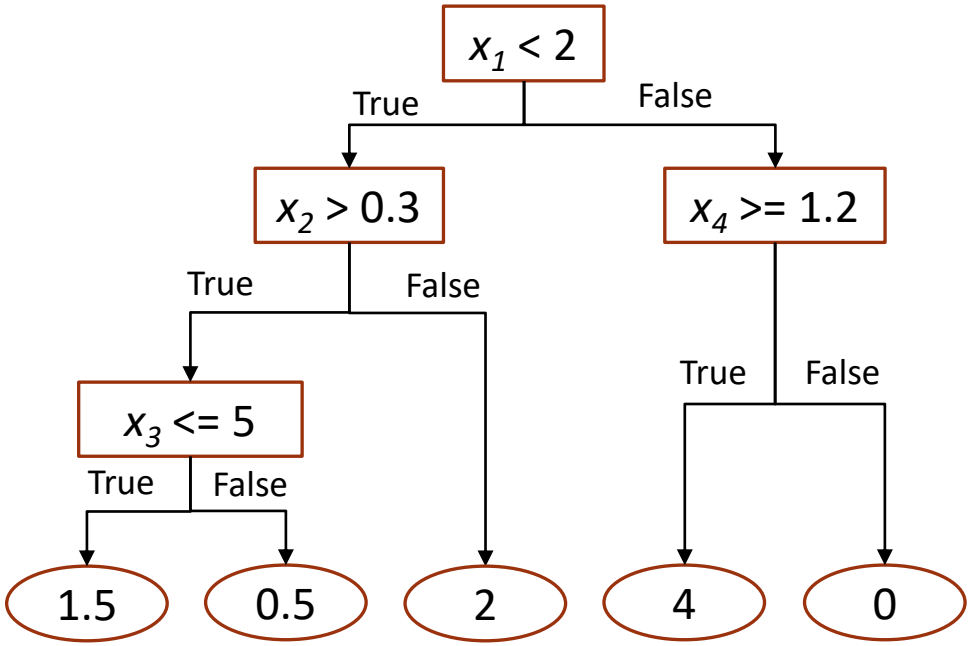


Árvores de Decisão

Também conhecidas como **Árvores de Regressão** são criadas por algoritmos que usam critérios como a minimização do quadrado dos erros.

Semelhantes às árvores de classificação, em vez de terem classes nas folhas, têm valores numéricos.

Sendo x_1, x_2, … x_n os atributos do nosso conjunto de dados, um exemplo de uma árvore de regressão seria:



Ensembles

Conjuntos de modelos individuais que são agregados de modo a obter-se uma resposta única.

Pode utilizar-se a média ou média pesada dos *outputs* dos modelos individuais para se obter um *output* único.

A maioria dos modelos de ensemble usados em classificação, têm também implementações para tarefas de regressão (ex.: ***Random Forest, Extreme Gradient Boosting (XGBoost), AdaBoost, Extremely Randomized Trees***).

Ferramentas:

- Python: https://scikit-learn.org/stable/modules/ensemble.html; https://xgboost.readthedocs.io/.
- R: https://daviddalpiaz.github.io/r4sl/ensemble-methods.html#regression-2.
- Rapidminer: https://towardsdatascience.com/how-to-create-ensemble-models-using-rapid-miner-72a12160fa51.
- Weka: https://machinelearningmastery.com/use-ensemble-machine-learning-algorithms-weka/.

Hiper-parâmetros

Os modelos de *Machine Learning* têm um conjunto de parâmetros que podem influenciar a sua aprendizagem. Exemplos:

- **Árvores de Decisão:** profundidade máxima (*max_depth),* influencia o seu crescimento.
- ***Random Forest:*** número de árvores *(n_estimators)*, por quantas árvores é composto.
- **Redes Neuronais:** número de camadas (arquitetura), otimizador (algoritmo de ajuste dos pesos),…

Como escolher os melhores valores?

- Usar os **valores padrão** (*default*) das implementações que usamos;
- Afinação destes valores por **tentativa-erro** (convém saber o que estamos a fazer!);
- Usar um **método de procura** para afinar estes valores por nós (ex.: *grid-search, automated search*, algoritmos genéticos,…).

Importante: não usar os dados de teste para tomar decisões!!!

CRISP-DM – Avaliação (regressão)

Avaliação de modelos deve ser feita de forma **objetiva** → uso de métricas para medir a qualidade das previsões.

2 tipos de avaliação:

- **Interna –** medida nos dados de **treino**.
- **Externa –** medida nos dados de **teste** (não utilizados no treino do modelo); serve para medir a capacidade de generalização dos modelos.

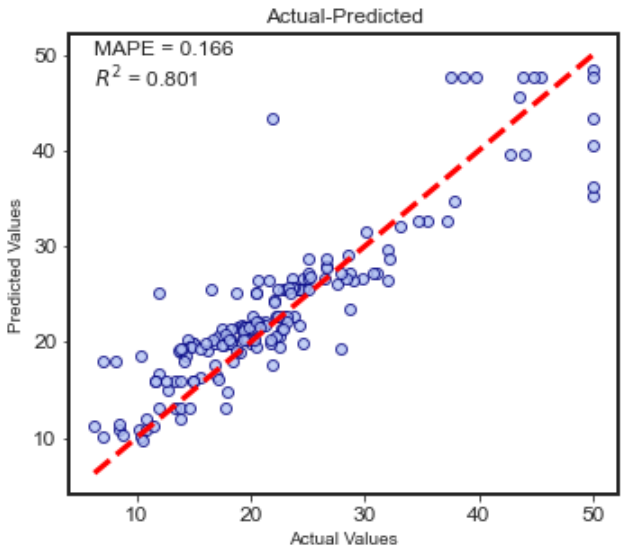
Métricas de regressão:

- **MAE –** mean absolute error (min., [0, Inf[).
- **RAE –** relative absolute error (min., [0%, Inf[).
- **NMAE -** normalized mean absolute error (min., [0%, Inf[).
- **SSE –** sum squared error (min., [0, Inf[).
- **MSE –** mean squared error (min., [0, Inf[).
- **RMSE –** root mean squared error (min., [0, Inf[).
- **RSE –** relative squared error (min., [0%, Inf[).
- **RRSE –** root relative squared error (min., [0%, Inf[).
- **R2 -** coefficient of determination (max.,]-Inf, 1]).
- **Tolerance –** the tolerance (y-axis value) of a REC curve (max., [0,1.0]).

CRISP-DM – Avaliação (regressão)

RSC (*Regression Scatter Plot*) – apresenta os **valores reais** (eixo-x) vs. **valores previstos** (eixo-y).

Quanto mais próximo os valores estiverem da linha vermelha, melhor!

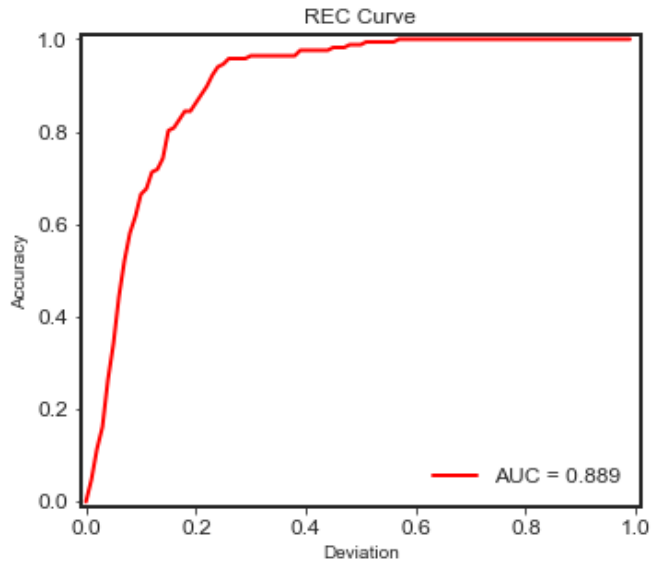


Curva REC (*Regression Error Characteristic*) – <u>não confundir com curva ROC</u> (classificação)! ⚠

Para uma determinada tolerância ao erro (eixo-x), apresenta a taxa de acerto (eixo-y).

Neste caso, *accuracy* corresponde à quantidade de valores que se encontram dentro da nossa tolerância ao erro.

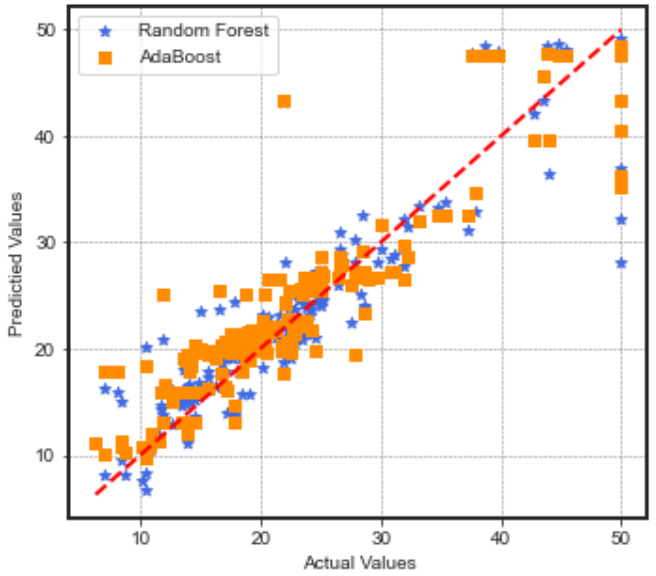
Quanto maior a área debaixo da curva REC (AUC), melhor!

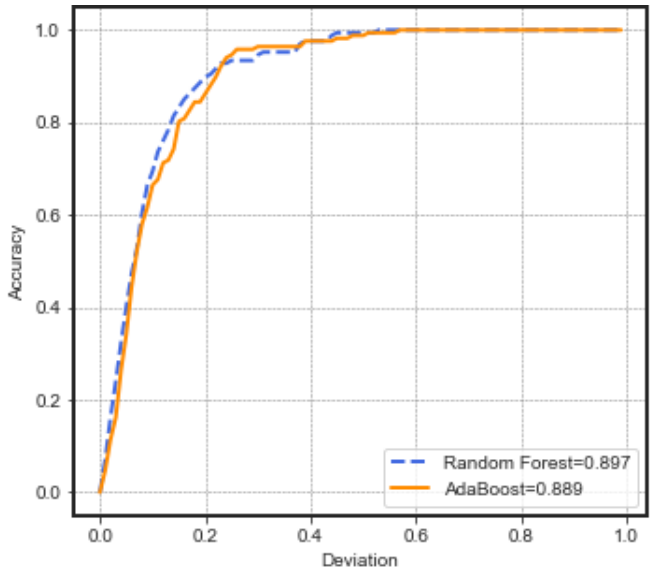


CRISP-DM – Avaliação (regressão)

RSC – interpretação **difícil** quando tentamos comparar vários modelos (os pontos podem sobrepor-se): não recomendado!

Curva REC – facilita a visualização quando comparamos vários modelos ou várias execuções.





Ferramentas:

- Python: https://scikit-learn.org/stable/modules/model_evaluation.html; https://github.com/amirhessam88/Regression-Error-Characteristic-Curve/blob/master/examples/example.ipynb.
- R: https://www.rdocumentation.org/packages/rminer/versions/1.4.6/topics/mmetric.
- Rapidminer: https://docs.rapidminer.com/latest/studio/operators/validation/performance/predictive/performance_regression.html .

Avaliação: validação de modelos

A validação de modelos pretende "estimar" a sua capacidade de generalização, medindo a sua qualidade/desempenho.

Como tal, as métricas **não podem ser calculadas utilizando dados que o modelo já "viu".**

Holdout: divisão dos dados em dois conjuntos exclusivos, através de uma amostragem aleatória.

- **Treino:** tipicamente **2/3 do conjunto dos dados**, usado para treinar modelos e tomar decisões (melhor modelo, melhores hiper-parâmetros, melhor pré-processamento,…). Por vezes, este conjunto é subdividido em 2 conjuntos (**treino** e **validação**) para verificar decisões internas do modelo.
- **Teste:** tipicamente **1/3 do conjunto dos dados**, é utilizado para avaliar as capacidades do modelo.

Ferramentas:

- Python: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- R: https://rdrr.io/cran/rminer/man/holdout.html
- Rapidminer: https://docs.rapidminer.com/latest/studio/operators/blending/examples/sampling/split_data.html

Aprendizagem Automática em Sistemas Empresariais

PEDRO PEREIRA

AULA 6