

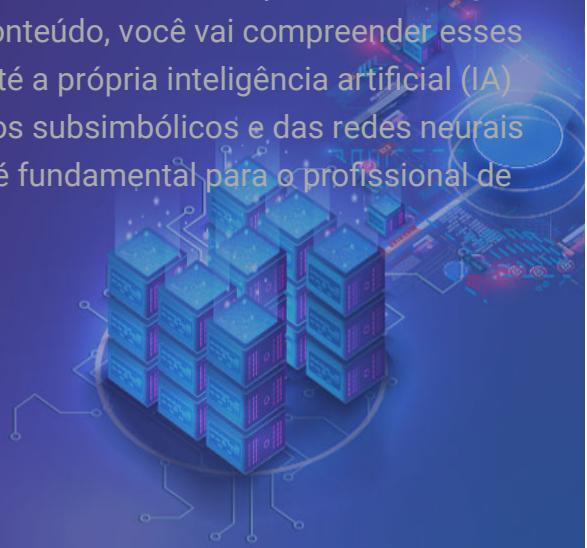
# Big Data Analytics

Os conceitos de Big Data Analytics têm ganhado popularidade com o avanço das tecnologias para tratamento de grandes volumes de dados. Neste conteúdo, você vai compreender esses conceitos e seus processos, desde a análise de dados até a própria inteligência artificial (IA) que a instrumentaliza, além da demonstração de modelos subsimbólicos e das redes neurais com TensorFlow e Python. Adquirir esse conhecimento é fundamental para o profissional de tecnologia da informação.

 Tempo total de leitura  
86 min.

Créditos

 Professor (a)  
**FERNANDO DURIER**



## Introdução

A área de dados está aquecida, demandando muito do mercado de tecnologia de informação e de áreas afins. Nesse contexto, passam a ganhar destaque o aprendizado de máquinas inteligentes e os benefícios que suas diversas aplicações podem trazer para a sociedade.

Há diferentes tipos de inteligência artificial (IA) que apresentam métodos e técnicas compatíveis com cada tipo de problema específico. Assim, podemos escolher melhor em qual estratégia apostar, além de aprender a implementar alguns desses modelos, usando bibliotecas existentes na linguagem de programação Python.

É sobre essas estratégias e esses modelos que vamos estudar aqui!

## Preparação

Para compreender e reproduzir os exemplos apresentados ao longo do conteúdo, você precisa instalar em seu computador a linguagem Python (na versão 3.8 ou superior) e as bibliotecas Pandas, Numpy, Sklearn e Plotly. Além disso, é necessário haver uma ferramenta para execução de notebooks, como Jupyter, que pode ser instalado como uma biblioteca do Python. Você também pode executar os códigos dos exemplos diretamente em um ambiente de execução, como, por exemplo, o Google Colab.

## Objetivos

Ao final desta aula, você será capaz de:

- Reconhecer o processo de KDD no contexto da inteligência artificial.
- Aplicar técnicas de aprendizado de máquina com Scikit-Learn.
- Reconhecer as técnicas de aprendizado profundo.
- Aplicar técnicas de aprendizado de máquina com Tensorflow.

# 01. Knowledge discovery in databases

## Inteligência artificial

### Evolução da internet

Atualmente, a área de ciência de dados tem ganhado muita popularidade em virtude dos resultados de análise de dados que são entregues de forma rápida, eficiente e com custo reduzido.

Para entendermos melhor isso, precisamos voltar um pouco no tempo. Vamos juntos? Veja agora.

#### Década de 1960

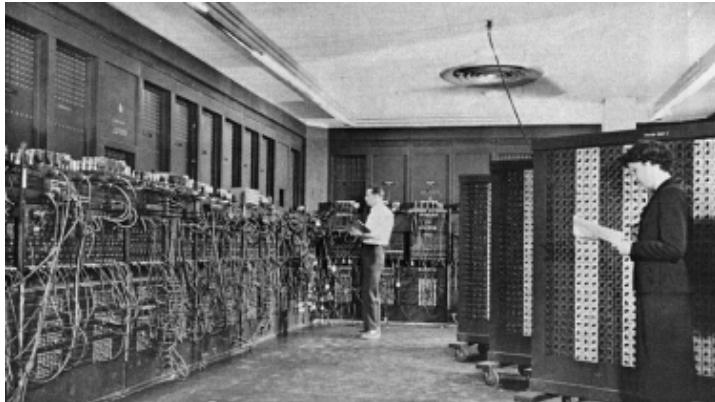
Tudo começou com a internet, que, a princípio, surgiu como uma pesquisa científica na área de defesa, ainda como ARPANET (Advanced Research Projects Agency Network), patrocinada pelo Departamento de Defesa do governo dos Estados Unidos.



#### Década de 1970

Com a disseminação de seu uso acadêmico por universidades envolvidas na pesquisa, a partir dos anos 1970 a internet começou

a ganhar os contornos que conhecemos hoje.



## Década de 1980

A internet foi impulsionada pela popularização dos computadores pessoais.



## Década de 1990

Com o advento da world wide web (www), a internet foi difundida na sociedade.



## Fases da web

Da década de 1990 até os dias atuais, a web evoluiu e passou por várias fases. Vamos conhecê-las? Confira a seguir.

### Web 1.0

Era caracterizada por ser a web do consumo de informações na rede. Os detentores da informação eram:



### Institutos governamentais



## Universidades



## Empresas de tecnologia vinculadas ao governo

O papel da Web 1.0 era prover informações relevantes sobre assuntos relacionados às suas áreas de interesse. Normalmente, os usuários consumiam os dados de fontes específicas de acesso, como bibliotecas, centros de pesquisa e gabinetes governamentais.

## Web 2.0

Iniciada em meados dos anos 2000, tal fase, denominada web semântica, já apresentava o computador pessoal como uma realidade nas tarefas domésticas da sociedade moderna.

## Curiosidade

Começava a se observar aí um fenômeno interessante: colaboração cibernética, surgimento das primeiras redes sociais de modo bem rudimentar (inicialmente em forma de blogs e fóruns, entre outros exemplos) e crescimento de aplicações de comércio eletrônico com recursos de relacionamento com o cliente (CRM). Os usuários passaram a contribuir também para a produção de dados nesse novo mundo digital.

Por volta de 2007, ocorreu uma mudança na forma de acesso à internet tão radical quanto a revolução da www nos anos 1990.



A tecnologia de hardware, que vinha crescendo com recursos computacionais cada vez mais potentes, deu um salto com a evolução dos aparelhos telefônicos móveis para os smartphones. Isso propiciou a geração e o consumo de um grande volume de dados com variedade de formatos, propagando-se em alta velocidade.

Esse grande fenômeno de produção e disseminação de dados é conhecido como Big Data e contava com três Vs: **volume, variedade e velocidade**. Esses dados eram gerados pela colaboração humana com o ciberespaço por meio da troca de informação entre os sistemas da web graças a dados produzidos pelos novos dispositivos pessoais conectados à rede.

## Web 3.0

Com o Big Data e o avanço tecnológico, entramos nessa fase da hiperpersonalização. Destacam-se aqui os algoritmos e motores de busca que tentam aproveitar essa massa de dados para extrair padrões e fazer recomendações aos usuários ou incorporá-las ao negócio de grandes organizações capazes de dominar essa tecnologia. Isso reaqueceu a área de inteligência artificial (IA), estimulando o surgimento dos conceitos de mineração e ciência de dados.



## O que é inteligência artificial?

A IA é uma área de pesquisa da computação dedicada a buscar métodos ou dispositivos computacionais que possuam ou multipliquem a capacidade racional do ser humano de resolver problemas, pensar ou ser inteligente. A máquina simula o comportamento humano por meio de:

- Processamento da linguagem natural;
- Visão computacional ou processamento de imagens;
- Representação do conhecimento;
- Raciocínio com lógica (de predicados, modal, difusa etc.);
- Sistemas de agentes.

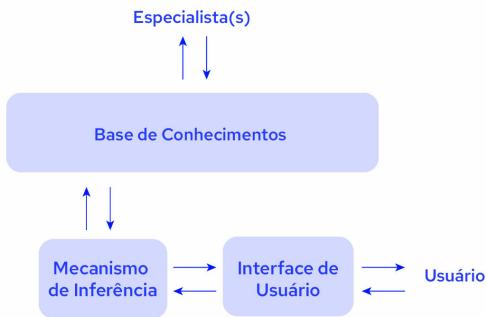
Definindo esses sistemas como agentes, verificamos que eles são capazes de aprender com o meio em que vivem mediante observações feitas por intermédio de sensores e ações tomadas por atuadores – e baseadas em processamentos lógicos que ocorrem graças ao aprendizado de máquina.

## Tipos de inteligência artificial

### IA simbólica

Segundo Wnek e Michalski (1992), o conhecimento representado por regras baseadas em lógica ou árvores de decisão é relativamente fácil de compreender e de fazer associações com o modelo de raciocínio lógico humano.

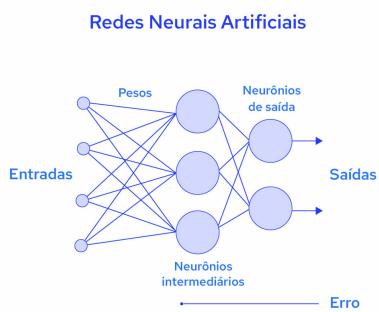
Essa é a abordagem simbólica, na qual problemas se transformam em fórmulas e expressões lógicas. O raciocínio ocorre a partir de operações lógicas feitas sobre símbolos contra axiomas e regras já conhecidos.



IA simbólica – sistemas de agentes.

## IA subsimbólica ou conexionista

Para essa abordagem, o conhecimento é representado a partir de modelos de sistemas classificadores ou redes neurais. A necessidade da IA subsimbólica ou conexionista surgiu da decepção com a IA simbólica, que não conseguia lidar bem com processamentos robustos e flexíveis, de acordo com Babbar e outros autores (2018), além do fato de ocultar o processamento de raciocínio de usuários não autorizados.



IA subsimbólica – redes neurais artificiais.

## Atividade

### Questão 1

O surgimento da web permitiu o acesso à internet a qualquer hora, em qualquer lugar e em qualquer dispositivo, possibilitando a troca de dados com inovação, naveabilidade e maior velocidade de informação. Isso estimulou o desenvolvimento da área de Big Data, que estuda como tratar, analisar e obter informações a partir de conjuntos de

dados grandes demais para serem analisados por sistemas tradicionais. Em qual fase da web nasceu o tema de Big Data?

A Web da informação.

B Web semântica.

C Web 1.0.

D Web 2.0.

E Web 3.0.

Parabéns! A alternativa E está correta.

A partir de 2007, o termo “Big Data” surgiu devido à produção de um grande volume variado de dados que se propagou de forma veloz com o desenvolvimento dos smartphones e de novos dispositivos conectados em redes.

# Aprendizado de máquina

## Métodos de aprendizado de máquina

O aprendizado de máquina é a forma como implementamos os processos cognitivos que a inteligência artificial (IA) tenta simular, pois, assim como nós temos diferentes processos cognitivos para determinadas tarefas do dia a dia, a IA também tem para a resolução de classes de problemas similares.

Os métodos de aprendizado de máquina conhecidos são:

### Aprendizado supervisionado

É o aprendizado no qual o modelo aprende a partir de exemplos positivos e negativos pré-rotulados para que a máquina possa mapear o padrão de entrada e saída esperado.

### Aprendizado não supervisionado

É o aprendizado autodidata dos modelos, ou seja, em vez de o modelo ter um conjunto pré-mapeado/rotulado, é apresentado a ele um conjunto sem marcações previas. Por meio de análise de critérios de similaridade, o modelo encontra por si só rótulos e padrões automaticamente.

### Aprendizado semissupervisionado

É o aprendizado no qual uma parte dos dados é rotulada previamente, enquanto outra é não rotulada. A maioria do conjunto é não rotulada. O modelo de aprendizado de máquina tenta, a partir do subconjunto menor, generalizar padrões para o conjunto maior ainda não rotulado.

# Aprendizado por reforço

É o aprendizado com o ambiente. O agente inteligente aprende políticas de ações com base em interações com o ambiente no qual ele é implementado a partir de recompensas ou punições, dependendo de cada ação.

## Técnicas de aprendizado de máquina

As técnicas mais populares são:

### Classificação

Técnica do aprendizado de máquina supervisionado que permite ao modelo conseguir entender como categorizar observações do conjunto de dados com base em registros históricos, bem como em suas características.

### Regressão

Contraparte numérica da classificação categórica. Também é uma técnica de aprendizado supervisionado em que o modelo aprende o mapeamento de entrada e saída para inferir um valor numérico em vez de uma classe categórica.

### Agrupamento

Técnica de aprendizado não supervisionado na qual o modelo, de forma autodidata, aprende a separar as observações, seguindo critérios de similaridades predefinidos, com o intuito de formar grupos de observações similares.

A IA não é uma área de pesquisa recente. Sua origem remonta aos anos 1950, antes mesmo do surgimento da internet como conhecemos. Mas, no princípio, a IA era rudimentar e não trazia, aos olhos das grandes organizações, tantos benefícios que justificassem investimento.

Graças ao advento da web (em especial, da web 2.0 e do *boom* de tecnologia de 2007), começou-se a produzir um volume de dados variados e em uma velocidade nunca vista de disseminação. Esses fatores impulsionaram a IA de uma forma que não era possível quando ela foi concebida inicialmente, dando a essa área força, popularidade e credibilidade.

Com mais pesquisas e avanços na tecnologia, além do surgimento de novos problemas contemporâneos oriundos do ciberespaço e da nova sociedade conectada, as pesquisas em IA avançam cada vez mais, evoluindo e mostrando agentes mais capazes e inteligentes.

**Não podemos esquecer a ciência de dados, que é a análise e a resolução de um problema de negócio (organizacional, industrial ou acadêmico) por meio das técnicas de aprendizado de máquina e de processamento de dados previstas na mineração de dados.**

A ciência de dados utiliza o instrumental do aprendizado de máquina para descobrir os padrões escondidos na grande massa de dados e dali tirar conclusões para problemas ocorridos em um negócio. Isso permite ao sistema se adaptar ao ambiente da organização graças à observação dos dados e à identificação de soluções que maximizem o ganho ou minimizem o custo. Com isso, a concretização da IA no mercado de trabalho praticamente vem se consolidando.

## Atividade

## Questão 1

Existe uma diferença, ainda que sutil, entre os conceitos de IA e de aprendizado de máquina. A IA se ocupa de simular agentes que aprendem com suas experiências e se adaptam a cenários de problemas em vez de dependerem de instruções pré-programadas. Já o aprendizado de máquina é a implementação dos processos cognitivos conhecidos para que as IAs operacionalizem seu dia a dia. Considerando essa diferença, aponte qual alternativa apresenta um exemplo de aprendizado de máquina.

- A Algoritmo genético.
- B Algoritmo Bubble Sort.
- C Algoritmo de redes neurais.
- D Assistente pessoal.
- E Web Crawler.

Parabéns! A alternativa C está correta.

O algoritmo de redes neurais implementa o processo cognitivo de representação do conhecimento a partir de ligação de impulsos; o algoritmo genético, um processo biológico de seleção natural para melhor resolução de problemas. Já o algoritmo Bubble Sort representa um processo de ordenação de comportamento pré-programado. O assistente pessoal é um exemplo de IA com um agente capaz de entender linguagem natural e devolver melhores resultados de recomendação. A Web Crawler, por fim, é não só um sistema de busca de links na web, mas também de comportamento tipicamente predefinido.

# Descoberta de conhecimento em bases de dados

## Conceitos básicos

Você sabe o que significa mineração de dados e como se encaixa na inteligência artificial (IA)?

Para entender o assunto, vamos conhecer alguns conceitos básicos a seguir:

### Dado

É um símbolo ou valor que, isoladamente, não significa muito.

Exemplo: sozinho, o dado 38 não tem muito sentido.

### Informação

É o dado processado, ou seja, que possui significado, aquele ao qual é feita uma atribuição.

Se dissermos, por exemplo, que 38 é a temperatura corporal de alguém, o dado do exemplo anterior começa a fazer sentido. E se tivéssemos a relação `idade=50, temperatura=38, gênero=masculino, peso=100kg e altura=1,70m?` Você deve ter percebido que tais informações são de um senhor de 50 anos acima do peso e com febre.

### Conhecimento

É o processamento de informação e a descoberta de padrões.

Imagine se, no exemplo anterior do senhor de 50 anos,

incluíssemos a variável de contexto

"esteve\_em\_contato\_com\_pessoas\_com\_COVID=sim"? Graças à análise de casos similares em uma base de dados, poderíamos deduzir que esse senhor está com coronavírus.

## Sabedoria

É o conhecimento processado que resulta em uma ação prática para resolver o problema. No caso do exemplo anterior, seria a resposta com uma prescrição médica relevante para supressão dos sintomas de covid e a recuperação do senhor.

## Sistema de informação

É um conjunto de partes que, individualmente, podem não ser dotadas de muitas funcionalidades, mas, em um sistema, trabalham em conjunto para atingir um objetivo final. A coleção de partes recebe dados como insumo, processa-os por meio de dinâmicas internas das partes e devolve informação relevante ao usuário final.

Exemplo: um sistema cadastra dados de pacientes e os salva em um banco de dados para outro sistema explorar e descobrir padrões.

## Mineração de dados

É um sistema de informação. O termo, porém, se confunde com o real nome do processo de descobrir padrões: *knowledge discovery in databases* (KDD) ou descoberta de conhecimento em bases de dados.

## Etapas do KDD

KDD é um processo que tem uma definição similar à de sistema: um conjunto de eventos ou partes – na maioria das vezes, subsequentes – que recebe um insumo e devolve um resultado processado dele.

O KDD pode ser dividido nas seguintes etapas:

### Coleta de dados

São capturados dados de uma organização para a resolução de um problema. Aqui temos o Big Data: uma massa de dados que traz não só bastante informação relevante, mas também muita irrelevante, podendo até atrapalhar a descoberta de padrões.

### Seleção

Assim como na operação de projeção em bases de dados, escolhemos quais características dessa grande massa de dados devemos considerar com base em sua relevância para o negócio.

### Pré-processamento

Esta é a etapa mais longa do processo; afinal, um cientista de dados passa 70% de seu tempo de trabalho nela.

No pré-processamento, entre outros exemplos, há limpeza nos dados, remoção de dados faltantes e corrompidos e incorporação de dados de outras bases.

## Transformação

O cientista regulariza a escala dos dados, reduz a dimensionalidade do conjunto e aplica outras técnicas de seleção de atributos automática. Isso é necessário para obter o conjunto de dados mais consistente e regular o máximo possível para que os algoritmos possam lidar com o mínimo de viés e ruído.

## Mineração de dados ou descoberta de padrões

O conjunto de dados é passado para um algoritmo de aprendizado de máquina compatível com o enquadramento do problema de dados, bem como da técnica em questão. No tópico aprendizado supervisionado (um dos métodos de aprendizado de máquina), isso significa que essa técnica estuda grande parte do conjunto e mede seus conhecimentos em outra parte. Feito isso, teremos um modelo de solução de nossos problemas.

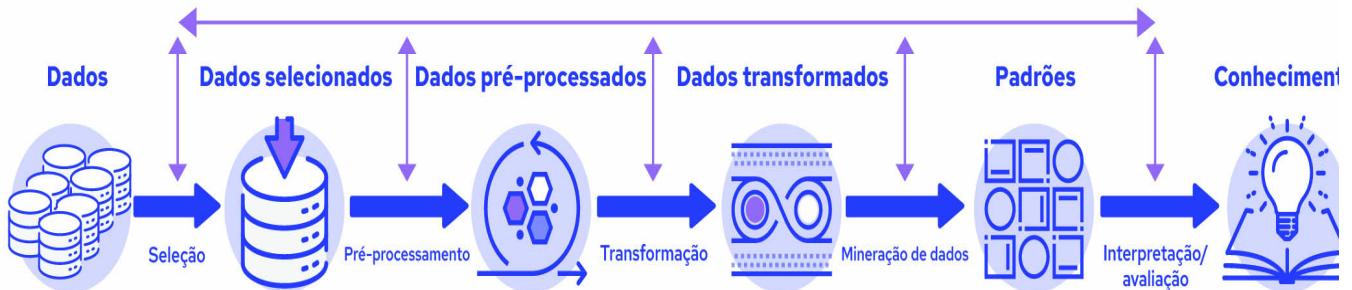
## Avaliação e apresentação de resultados

Entendemos quais padrões o modelo do algoritmo escolhido conseguiu aprender e com qual confiança. Avaliamos se esse modelo é confiável e se, de fato, aprendeu, se apenas decorou ou se simplesmente não aprendeu nada. Fazemos isso por meio das métricas, que mostram o comportamento do modelo, e das visualizações, que nos permitem ver o impacto mais concreto para o negócio.

Ao final do processo de KDD, obtemos o que precisávamos: um padrão descoberto a partir de dados que provêm informações. Isso resulta em conhecimento, o qual pode

ser utilizado por outros sistemas para resolver problemas, tornando-se, assim, sabedoria.

A imagem a seguir resume esse processo:



Processo de KDD.

## Atividade

### Questão 1

Quando falamos do trinômio dados, informação e conhecimento, estamos nos referindo ao conceito fundamental de sistemas de informação. No processo de KDD, qual dos componentes desse trinômio estamos tentando alcançar e em qual etapa o obtemos?

- A Dados – seleção.
- B Conhecimento – seleção.
- C Informação – transformação.
- D Conhecimento – interpretação e avaliação.

E

Conhecimento – pré-processamento.

Parabéns! A alternativa D está correta.

Após a avaliação dos resultados de padrões descobertos, podemos consolidar o conhecimento, que nada mais é do que informação processada durante todo o KDD. Os dados e a informação são aspectos anteriores ao conhecimento. Já as etapas de seleção e de pré-processamento são muito anteriores à geração de conhecimento.

# Processo CRISP-DM

## Etapas do CRISP-DM

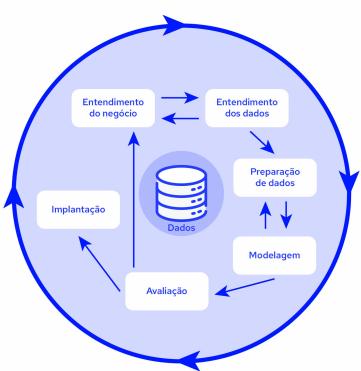
O processo de KDD tem um irmão que muitos pesquisadores consideram um subconjunto dele: o CRISP-DM (*cross industry standard process for data mining*).

Do mesmo modo que o KDD, o processo CRISP-DM possui seis etapas, mas elas têm outro foco. Listaremos essas etapas a seguir:

- Entendimento do negócio;
- Entendimento dos dados;
- Preparação de dados;
- Modelagem;
- Avaliação;
- Implantação.

Como você pode observar, não só o número de etapas é bem parecido, mas também os nomes de suas fases. E isso não é coincidência!

O processo do CRISP-DM é também conhecido como o KDD da indústria, pois tem um foco maior no negócio (tanto no começo quanto no fim). A imagem adiante resume esse processo:



Processo CRISP-DM.

---

Vejamos agora as semelhanças entre CRISP-DM e KDD, identificando as equivalências

entre as etapas (que têm as mesmas operações de suas contrapartes):

CRISP-DM	KDD
Preparação de dados	Selação, pré-processadores e transformação
Modelagem	Mineração de dados
Avaliação	Avaliação e apresentação de dados

Tabela: Comparação entre os processos CRISP-DM e KDD.

Fernando Durier

## CRISP-DM x KDD

Vamos estabelecer a seguir as diferenças entre os dois processos.

O modelo CRISP-DM traz o **entendimento do negócio e o entendimento dos dados no lugar da coleta de dados**. Esse modelo também oferece a **implantação (deployment)** como atividade seguinte à avaliação – e isso é um diferencial importante.

### CRISP-DM

Tem como entrega de valor final o artefato do modelo de aprendizado de máquina, por ser um processo voltado para a indústria.



### KDD

Não tem a obrigação de entregar um artefato, e sim o conhecimento apenas, que já é suficiente.

O modelo CRISP-DM será capaz de resolver os problemas propostos pelo negócio na

primeira etapa.

Esta é a maior diferença entre os dois processos: **a entrega de valor final**. Por isso, no fim das contas, a indústria prefere o CRISP-DM, enquanto a academia preza pelo KDD.

Lembre-se de que esse é um comportamento esperado, mas não é uma obrigatoriedade. Afinal, se a organização se contenta com a descoberta de padrão para um nível estratégico, o uso do KDD é suficiente. Em contrapartida, ao adquirir conhecimento sobre o KDD, aprende-se o fundamento, enquanto o CRISP-DM pode ser encarado como uma instanciação do KDD na indústria.

## Atividade

### Questão 1

Como os processos de CRISP-DM e de KDD apresentam etapas muito parecidas, é quase possível usar um no lugar do outro. Afinal, ambos recebem dados, os tratam de alguma forma, os transformam em um formato compatível e os repassam aos modelos de aprendizado de máquina para que esses modelos possam descobrir padrões.

O que diferencia tais processos é um detalhe simples: sua entrega de valor.

Considerando essa distinção, assinale a alternativa que apresenta respectivamente as entregas de valor do processo de KDD e do CRISP-DM.

A Artefato de aprendizado de máquina e conhecimento.

B Artefato de aprendizado de máquina e dados.

C Dados e conhecimento.

D Informação e dados.

E

Conhecimento e artefato de aprendizado de máquina.

Parabéns! A alternativa E está correta.

O processo de KDD está preocupado em entregar um padrão descoberto, uma visualização de dados que mostre o agrupamento de classes ou até mesmo uma regra de negócio recém-descoberta, podendo resultar em um sistema ou não. Já o CRISP-DM, de fato, precisa entregar um artefato de aprendizado de máquina, ou seja, um algoritmo de classificação, regressão ou agrupamento que receba informações da organização e devolva para o usuário um rótulo (positivo ou negativo), um valor ou um agrupamento. Essa solução pode ser só o algoritmo hospedado em algum servidor para ser consumido via API (interface de programação de aplicações) ou um sistema completo com interface e tudo.

# 02. Aprendizado de máquina com Scikit-Learn

## Classificação com SVM usando Scikit-Learn

### Implementando um classificador SVM em Python

1. Para implementar um classificador SVM em Python, instale as bibliotecas Pandas, Numpy, Matplotlib e Scikit-Learn. Comece com a importação das bibliotecas. Para isso, faça o seguinte:

Python



```
1  
2 import pandas as pd  
3 import numpy as np  
4 from matplotlib import pyplot as plt
```

2. Importe o dataset que será explorado em nossos estudos: o dataset Íris, considerado clássico no âmbito do aprendizado de máquina. Ele classifica tipos de flores da espécie íris a partir de características, como, por exemplo, comprimento e largura da pétala e comprimento e largura da sépala (todas em centímetros). Para tal, observando o destaque da coluna de classe (target) das características, adicione isto ao código:

Python



```
1  
2 from sklearn.datasets import load_iris  
3 data = load_iris()  
4 iris = pd.DataFrame(data['data'], columns=data.feature_names)  
5 target = data.target
```

3. Para a instanciação simples do classificador, observando que o classificador SVM é identificado no Scikit-Learn a partir do nome SVC (C de Classifier), adicione o seguinte bloco de código:

Python



```
1  
2 #Importando o algoritmo de SVM  
3 from sklearn.model_selection import cross_val_score  
4 from sklearn.svm import SVC  
5 svc = SVC(gamma="auto")
```

4. Para treinar o modelo e conhecer sua performance, adicione o bloco de código a seguir. A validação cruzada é um dos modos mais comuns de treinar nossos modelos, pois ela divide o conjunto de dados em  $(k-1)/k$  partições de treinamento e  $1/k$  de teste de maneira circular e interativa. Com isso, há todas as  $1/k$  possíveis partições, que podem ser testadas contra o resto.

Python



```
1  
2 #Testando o modelo 'svc' na nossa base 'iris'  
3 cv_result = cross_val_score(svc, iris, target, cv=10, scoring='accuracy')  
4 #Retorna a acurácia em porcentagem do nosso modelo  
5 print('Acurácia com cross validation:', cv_result.mean()*100)
```

5. Agora que você experimentou construir o classificador com os parâmetros padrões da biblioteca, faça as predições. Para isso, treine nosso modelo com o dataset inteiro e tente predizer um valor inédito da seguinte forma:

Python



```
1  
2 svc.fit(iris, target)  
3 #Prediz a que classe pertencerá a flor com sépala de comprimento 6.9cm
```

```
4 svc.predict([[6.9,2.8,6.1,2.3]])
```

6. Visualize nossos dados e os hiperplanos definidos pelo modelo. Esses dados têm o seguinte comportamento:

Python



```
1  
2 plt.scatter(iris['sepal length (cm)'], iris['petal width (cm)'], c=targ  
3 plt.title('Iris')
```

O resultado gráfico será:

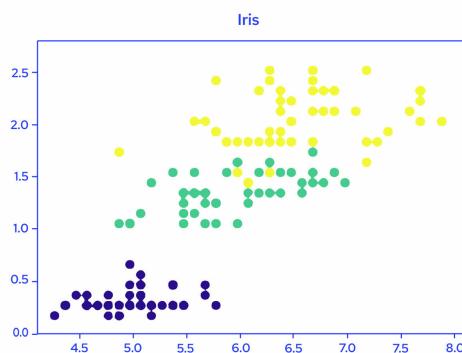


Gráfico: Dispersão dos dados íris.

Elaborado por: Fernando Durier.

7. Como você pode ver no gráfico, nosso problema parece ser linearmente separável. Faça isso com este outro bloco de código:

Python



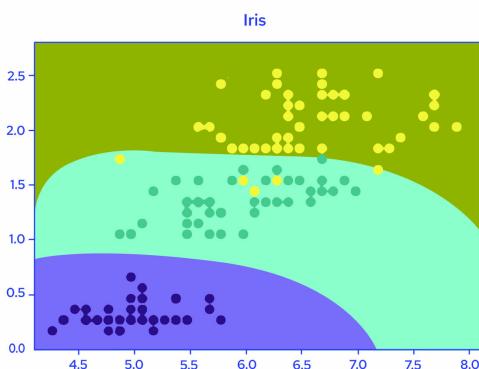
```
1  
2 #Provavelmente, ao criar 2 features novas no íris, o svm com 2 featur  
3 #sepal length e petal width (os mais relevantes das 4 features já exi  
4 x0_min, x0_max = iris['sepal length (cm)'].min(), iris['sepal length
```

```

5 x1_min, x1_max = iris['petal width (cm)'].min(), iris['petal width (c
6 w = x0_max - x0_min
7 h = x1_max - x1_min
8 x0, x1 = np.meshgrid(np.linspace(x0_min-.1*w, x0_max+.1*w, 300), np.l
9 svc.fit(iris[['sepal length (cm)', 'petal width (cm)']], target)
10 ypred = svc.predict(np.c_[x0.reshape(-1, 1), x1.reshape(-1, 1)])
11 ypred = ypred.reshape(x0.shape)
12 plt.contourf(x0, x1, ypred)
13 plt.scatter(iris['sepal length (cm)'], iris['petal width (cm)'], c=ta
14 plt.title('Iris')

```

O resultado será:



Hiperplanos gerados pelo modelo SVM.

Elaborado por: Fernando Durier.

Note que, de fato, o SVM foi capaz de definir os hiperplanos. Se você observar bem os hiperplanos superior e central desse gráfico, verá alguns outliers entre eles. Nesse caso, o algoritmo SVM os ignorou estrategicamente para não superajustar sua classificação, conferindo, assim, certa generalização ao classificador.

## Atividade

### Questão 1

Usando o mesmo conjunto de dados da prática anterior (de flores íris), investigue no Scikit-Learn o modelo de árvore de decisão, implemente seguindo o exemplo de documentação e imprima a métrica extraída pelo método score.

Você deve usar a classe DecisionTreeClassifier da biblioteca Sklearn. Ao executar o treinamento e a classificação, você obterá um score maior de 75% (0.75).

## Classificação com árvore de decisão usando Scikit-Learn

### Implementando uma classificação com árvore de decisão

1. Comece o código importando as bibliotecas Sklearn e Matplotlib e as funções necessárias:

Python



```
1  
2 import matplotlib.pyplot as plt  
3 from sklearn.datasets import load_iris  
4 from sklearn.model_selection import cross_val_score  
5 from sklearn.tree import DecisionTreeClassifier, plot_tree
```

2. Para realizar o processo de experimentação por meio do treinamento feito com validação cruzada, adicione este bloco de código:

Python



```
1  
2 clf = DecisionTreeClassifier(max_depth=3, random_state=0)  
3 iris = load_iris()  
4 cross_val_score(clf, iris.data, iris.target, cv=10)
```

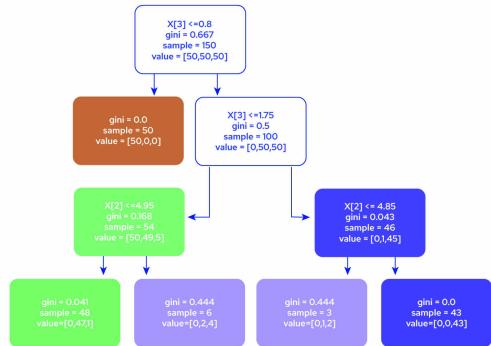
3. Com os dados carregados e o experimento executado para checar a possível média de acurácia do modelo treinado com o conjunto, acompanhe o treinamento propriamente dito e a visualização do resultado:

Python



```
1
2 clf.fit(iris.data, iris.target)
3 plot_tree(clf, filled=True)
4 plt.show()
```

Pronto! Com poucas linhas de código, você treinou um classificador de árvore de decisão. Veja sua árvore resultante.



Árvore de decisão resultante da implementação em Python.

Fernando Durier.

## Atividade

### Questão 1

Com o mesmo conjunto da prática anterior (de flores íris), investigue a documentação da biblioteca Scikit-Learn e utilize a classe de algoritmos de classificação GaussianNaiveBayes para fazer o treino e a classificação das flores do conjunto.

Primeiramente, você deve usar a classe GaussianNaiveBayes (GNB) da biblioteca Sklearn. Feito isso, ao executar o comando score, o valor deve ser maior do que 70% (0.7) e tem de ser comparado à árvore de decisão do exemplo do roteiro de prática. Com isso, você vai entender como os modelos se comportam e tomar sempre o GNB como uma *baseline* inicial para servir de termômetro. Se o modelo testado é pior em performance que o GNB, pode ser que o modelo comparado esteja mal configurado.

## 03. Aplicações de aprendizado profundo

### Redes neurais artificiais

#### O que é aprendizado profundo?

Do inglês *deep learning*, o aprendizado profundo é uma das técnicas que podem ser usadas na etapa de modelagem do CRISP-DM. Essa técnica corresponde à mineração de dados, o coração do processo de KDD.

O aprendizado profundo tenta modelar abstrações de alto nível de dados usando um grafo com várias camadas de processamento compostas por muitas transformações lineares e não lineares. Ele é um modelo de aprendizado de máquina que decompõe dados complexos em representações mais bem compreendidas pela máquina.

#### Exemplo

Vetores, matrizes e séries temporais.

Uma de suas maiores vantagens é incorporar nas camadas escondidas e profundas todo o pré-processamento e a geração de características, os quais, em modelos clássicos, são feitos em etapas anteriores e externas ao modelo. Devido a esse aspecto de múltiplas transformações lineares e de aprendizado conexionista, os grafos empregados nessa área são as **redes neurais artificiais**.

## Modelo de redes neurais

Assim como as contrapartes biológicas (neurônios), o modelo artificial tenta, por meio do ajuste iterativo de pesos das camadas da rede e das operações extras entre transições de camadas, representar os dados recebidos da maneira como nossos neurônios e nossa rede nervosa representam nossas observações, sensações e reações no mundo real. Cada camada da rede neural é um aspecto da observação, uma abstração do todo.

## Exemplo

Em redes que reconhecem imagens, cada camada pode representar um aspecto da imagem: a primeira pode detectar linhas; a segunda, círculos; e a terceira, composições entre linhas e círculos.

Já vimos que, na área de inteligência artificial (IA), existem dois grandes grupos:

- IA simbólica;
- Subsimbólica.

Basicamente, o que os difere é o processo de aprendizado. Ainda que ambos se enquadrem em aprendizado supervisionado e não supervisionado, o tipo de processo varia. Vejamos mais detalhes:

### Modelos simbólicos

O aprendizado é baseado em regras, propriedades e lógica.

×

### Modelos subsimbólicos

Há um aprendizado direto dos dados ou da experiência.

Esses modelos não precisam de etapas de pré-processamento para funcionar, pois isso já está embutido em seu funcionamento.

As redes neurais pertencem ao grupo de IA subsimbólica. Um dos primeiros modelos desse grupo foi o perceptron, que simulava um neurônio. Assim como sua contraparte biológica, ele tinha entradas e saídas por onde os estímulos passavam e uma função de ativação que levava entradas a um domínio desejado (segundo o conjunto de dados).

Individualmente, cada perceptron pode ser encarado como um classificador linear. O conjunto desses neurônios artificiais é uma rede neural.

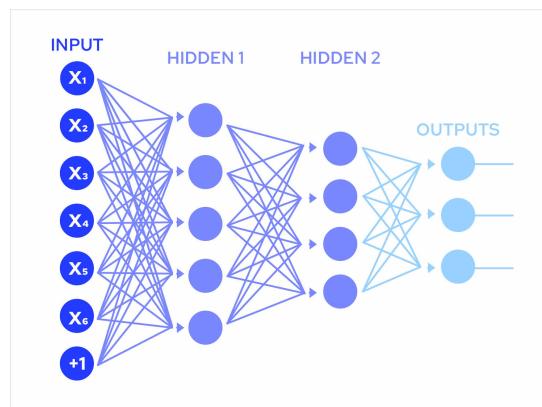
## Tipos de redes neurais

A área de redes neurais artificiais é muito vasta. Desse modo, ela sempre está caminhando para novas evoluções e variações de modelos.

Vejamos alguns tipos de redes neurais.

## Redes neurais convolucionais

*Convolutional neural network* (CNN) são muito utilizadas em processamento de imagens. Sua arquitetura é composta de camadas de entrada e saída e de camadas escondidas ou ocultas, diferenciando-se dos demais tipos pelo fato de que contam com filtros – em geral, matrizes que deslizam sobre a entrada de dados.



As camadas ocultas mapeiam de forma equivariante os pesos e os filtros para as camadas subsequentes, sendo invariantes ao espaço e ao deslocamento. A maior vantagem desse tipo de rede neural é sua maior autonomia.

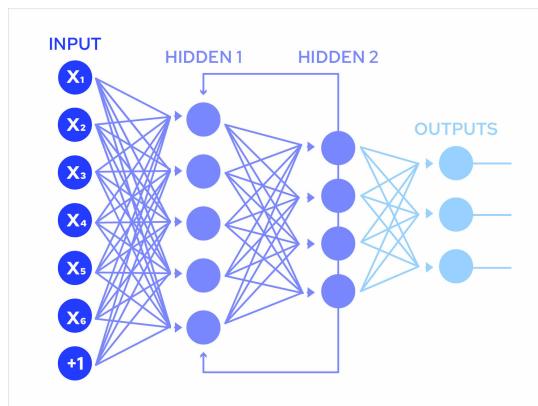
## Exemplo

Para processamentos de imagem, a rede não precisa ter filtros preestabelecidos pelo cientista de dados, já que ela consegue apreender o melhor núcleo de filtragem (matriz

de filtragem).

## Redes neurais recorrentes

*Recurrent neural networks* (RNN) são usadas em problemas sequenciais ou temporais. Como sua estrutura é a de um grafo direcionado com ciclos, essa rede é capaz de recorrentemente realizar laço(s) sobre as entradas, permitindo a compreensão de sequenciamento dos dados. A arquitetura mais conhecida dessa categoria é a LSTM (*long short-term memory*).



De modo superficial, o que diferencia essas redes de seus concorrentes é o fato de elas terem portas (*gates*) de esquecimento, permitindo à rede não exagerar nos pesos nem para mais, nem para menos. Em redes RNN comuns, o gradiente (base do aprendizado desse modelo) pode acabar sumindo à medida que se propaga para trás ou aumentando ao se propagar para frente.

Para evitar esse problema computacional de exageros nos gradientes, esse tipo de rede apresenta unidades de memória de curto prazo (*short term memory*), que servem para lembrar os estados anteriores, evitando que o gradiente suma. Além disso, tais redes podem ser reguladas com pequenos episódios de perda de memória desejada para não exacerbar os gradientes.

## Atividade

### Questão 1

Aprendizado profundo é uma técnica de Machine Learning que ensina aos computadores o que nós, humanos, fazemos de maneira fácil e natural: aprender com

os exemplos encontrados. Nesse contexto, o que as camadas escondidas representam em um modelo de aprendizado profundo?

A Operações

B Relações

C Números

D Abstrações

E Categorias

Parabéns! A alternativa D está correta.

Os modelos de aprendizado profundo tentam incorporar o pré-processamento de dados para que, justamente no cerne dessa estrutura (as camadas ocultas), o modelo seja capaz de capturar diferentes aspectos dos dados complexos ou das abstrações.

# 04. Aprendizado de máquina com TensorFlow

## Demonstração de redes neurais usando TensorFlow

### Implementando uma rede neural usando TensorFlow

1. Instale o TensorFlow usando o código: `pip install --upgrade tensorflow`

Lembre-se de que o TensorFlow pode apresentar restrições por sistema operacional. Por isso, tome cuidado e preste atenção nos logs do terminal, pois eles podem indicar instruções alternativas de instalação. Em último caso, basta criar um notebook compartilhado no Google Drive e fazer a experimentação pelo Google Colab.

2. Instale as bibliotecas de ciência de dados que darão suporte a métricas e gráficos, como o Scikit-Learn, Numpy e o Matplotlib:

```
pip install sklearn numpy matplotlib
```

3. Para finalizar a preparação, instale o Keras, que é uma biblioteca para o desenvolvimento de redes neurais projetada para permitir experimentação rápida. Para isso, digite o seguinte código:

```
pip install keras
```

4. Importe as bibliotecas necessárias à demonstração:

Python



```
1
2 from sklearn.preprocessing import LabelBinarizer
3 from sklearn.metrics import classification_report
4 from tensorflow.keras.models import Sequential
5 from tensorflow.keras.layers import Dense
```

```
6 from tensorflow.keras.optimizers import SGD  
7 from tensorflow.keras.datasets import mnist  
8 from tensorflow.keras import backend as K  
9 import matplotlib.pyplot as plt  
10 import numpy as np
```

## 5. Baixe o conjunto de dados para experimentação.

Utilizaremos um dos conjuntos de dados mais conhecidos e usados na área de redes neurais: os conjuntos de dígitos escritos à mão do MNIST (Modified National Institute of Standards and Technology). Basicamente, nossa rede neural vai aprender a reconhecer o padrão de escrita de números.

Para conseguir esse conjunto, utilize o seguinte bloco de código:

```
print("[INFO] accessing MNIST...")  
((trainX, trainY), (testX, testY)) = mnist.load_data()
```

## 6. Uma vez baixado, o conjunto já vem no formato de treinamento e teste, como podemos ver na declaração do bloco de código anterior. No entanto, você vai precisar rearrumar o conjunto.

Cada imagem do MNIST tem as dimensões 28x28x1. Porém, para a rede neural, chape a imagem em  $28 \times 28 = 784$  pixels.

## 7. Em seguida, normalize os dados para que eles fiquem entre 0 e 1. Faça isso dividindo o conjunto por 255 (valor máximo de um pixel). Para tal, utilize este código:

Python



```
1
```

```
2 trainX = trainX.reshape((trainX.shape[0], 28 * 28 * 1))
```

```
3 testX = testX.reshape((testX.shape[0], 28 * 28 * 1))
4 trainX = trainX.astype("float32") / 255.0
5 testX = testX.astype("float32") / 255.0
```

8. Para adequar a última camada (a de saída), binarize a classe da seguinte forma:

Python



```
1
2 lb = LabelBinarizer()
3 trainY = lb.fit_transform(trainY)
4 testY = lb.transform(testY)
```

O LabelBinarizer faz com o que o resultado da classe se torne binário, ou seja, em vez de lidarmos com a classe de valor 5, passaremos a fazê-lo com 0000100000. Isso se mostra importante, já que a camada final deve ter um tamanho proporcional às possibilidades de resultados esperados.

9. Defina a arquitetura da nossa rede neural. Com a ajuda do Keras, isso pode ser feito de maneira simples, adicionando uma camada atrás da outra em sequência, como podemos ver a seguir:

Python



```
1
2 model = Sequential()
3 model.add(Dense(256, input_shape=(784,), activation="sigmoid"))
4 model.add(Dense(128, activation="sigmoid"))
5 model.add(Dense(10, activation="softmax"))
```

10. Para treinar o modelo, use como otimizador do gradiente o SGD, aquele baseado no gradiente descendente e cuja taxa de aprendizado é 0.01. Também utilize a métrica de acurácia para acompanhar o modelo.

Para calcular a perda ou a função de custo, empregue a entropia cruzada categórica (categorical\_crossentropy), que é a mais utilizada em problemas de classificação.

Para as épocas da nossa rede, escolha 100 épocas, ou seja, a rede tem 100 iterações para convergir e apreender; em seguida, apresente lotes de 128 imagens cada por iteração.

Para isso, digite o código adiante:

Python



```
1
2 sgd = SGD(0.01)
3 model.compile(loss="categorical_crossentropy", optimizer=sgd, metrics=["
4 H = model.fit(trainX, trainY, validation_data=(testX, testY), epochs=100
```

11. Agora chegou o momento de ver como a rede se saiu. Para isso, utilize a classification\_report, uma função do Sklearn que compara os valores preditos com os reais, que são passados como argumentos. Basta digitar este bloco de código:

Python



```
1
2 predictions = model.predict(testX, batch_size=128)
3 print(classification_report(testY.argmax(axis=1),
4 predictions.argmax(axis=1),
5 target_names=[str(x) for x in lb.classes_]))
6
```

O resultado disso será um relatório de classificação. Como esse problema é multiclasse, além de mostrar a acurácia geral da classificação, o relatório apresentará o resultado de cada classe possível.

12. Por fim, veja como a rede evoluiu até chegar a essas métricas, ou seja, como a função de custo foi otimizada e a acurácia subiu. Para isso, utilize o seguinte bloco de código:

Python



```
1
2 plt.style.use("ggplot")
3 plt.figure()
4 plt.plot(np.arange(0, 100), H.history["loss"], label="train_loss")
5 plt.plot(np.arange(0, 100), H.history["val_loss"], label="val_loss")
6 plt.plot(np.arange(0, 100), H.history["accuracy"], label="train_acc")
7 plt.plot(np.arange(0, 100), H.history["val_accuracy"], label="val_acc")
8 plt.title("Training Loss and Accuracy")
9 plt.xlabel("Epoch #")
10 plt.ylabel("Loss/Accuracy")
11 plt.legend()
12
```

Esse código resultará no gráfico adiante. Como podemos ver, o resultado da função de custo diminui à medida que as épocas passam, enquanto a acurácia aumenta, o que é bastante intuitivo, uma vez que a rede está aprendendo com o passar das épocas (iterações).



Gráfico: Curva resultante do código anterior.

# Atividade

## Questão 1

Agora que você já sabe como configurar uma rede neural inicial, é sua vez de experimentar!

Com a mesma rede neural configurada na prática anterior e o mesmo conjunto de dados (de números MNIST), altere as propriedades da rede. Adicione mais camadas ou remova as existentes. Mude as funções de ativação e altere os valores de batch\_size e epochs.

A cada alteração, execute o treinamento e aguarde. Anote os valores em uma planilha e faça novas alterações. Pare quando chegar à conclusão de que houve alguma melhoria nas métricas.

[Abrir solução ▾](#)

Trata-se de um processo de experimentação de modelos (ou hyper parameter tuning) na força bruta que nos permite entender como a alteração de cada parâmetro mexe no resultado final do classificador. O indicativo de que o exercício está correto é não dar erro no motor do Python. Você deve alterar os valores e anotar os efeitos para escolher a melhor configuração.

# Conclusão

O que você aprendeu neste conteúdo?

- Processo de KDD no contexto da inteligência artificial;
- Técnicas de aprendizado de máquina com Scikit-Learn;
- Técnicas de aprendizado profundo;
- Técnicas de aprendizado de máquina com TensorFlow.

Explore +

Uma excelente introdução (em inglês) sobre Data Mining e Machine Learning pode ser lida no livro on-line **Data Mining and Machine Learning: fundamental concepts and algorithms**, publicado pelos professores Mohammed Zaki e Wagner Meira Jr.

Conheça outras funcionalidades da **biblioteca Scikit-Learn** no site do próprio projeto.

## Referências bibliográficas

AMARAL, F. **Aprenda mineração de dados: teoria e prática.** v. 1. Rio de Janeiro: Alta Books, 2016.

AZEVEDO, A. I. R. L; SANTOS, M. F. KDD, SEMMA and CRISP-DM. **A parallel overview.** IADS-DM, 2008.

BABBAR, P. **Et al: International journal of applied engineering research.** v. 13. n. 7. 2018. p. 5154-5159.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. **The KDD process for extracting useful knowledge from volumes of data.** Communications of the ACM. v. 39. n. 11. 1996. p. 27-34.

KABIR, S. M. S. **Methods of data collection.** In: KABIR, S. M. S. Basic guidelines for research – an introductory approach for all disciplines. 1. ed. Chatigão: Book Zone, 2016. cap. 9. p. 201-

RUSSEL, S.; NORVIG, P. **Inteligência artificial**. 3. ed. São Paulo: GEN LTC, 2013.

SILVA, F. C. D.; GARCIA, A. C. B. **Judice verum**. A methodology for automatically classify fake, sarcastic and true portuguese news. dez. 2019.

WIRTH, R.; HIPP, J. CRISP-DM. **Towards a standard process model for data mining**.

Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining. Springer-Verlag, 2000.

WNEK, J.; MICHALSKI, R. S. **Comparing symbolic and subsymbolic learning: three studies**.

Airfax: George Mason University, 1992.