Análise de dados com foco em Machine Learning usando o R e o Python

Paulo Henrique Sales Guimarães

Departamento de Estatística - UFLA

Conteúdo do curso

Conteúdo do curso

Data Science

Conteúdo do curso

Data Science

Manipulação e análise de dados

Conteúdo do curso

Data Science

Manipulação e análise de dados

Machine Learning

Conteúdo do curso

Data Science

Manipulação e análise de dados

Machine Learning

Um pouco sobre linguagem R e Python

Conteúdo do curso

Data Science

Manipulação e análise de dados

Machine Learning

Um pouco sobre linguagem R e Python

R e Python juntos?

Data Science

O que é Data Science?

Data Science

O que é Data Science?

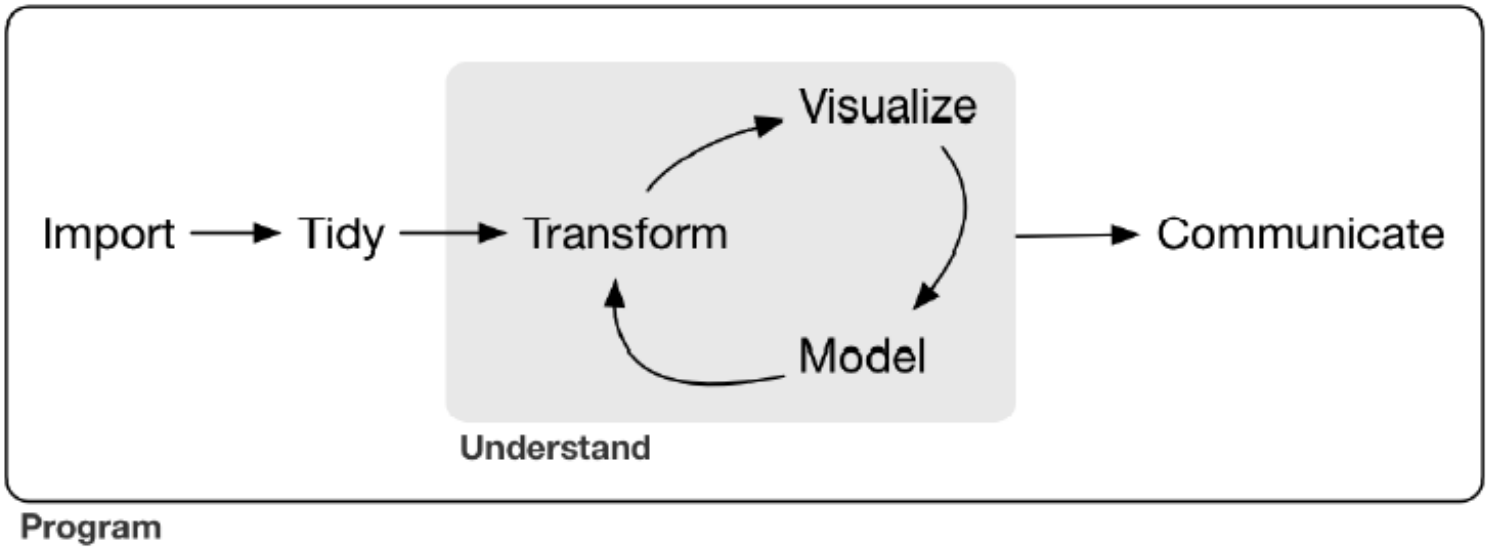


Figura : R for Data Science - Hadley Wickham & Garrett Grolemund.

Data Science

O que é Data Science?

Data Science

O que é Data Science?

Figura : Disponível em: https://towardsdatascience.com/.

Data Science

O que é Data Science?

Data Science

O que é Data Science?

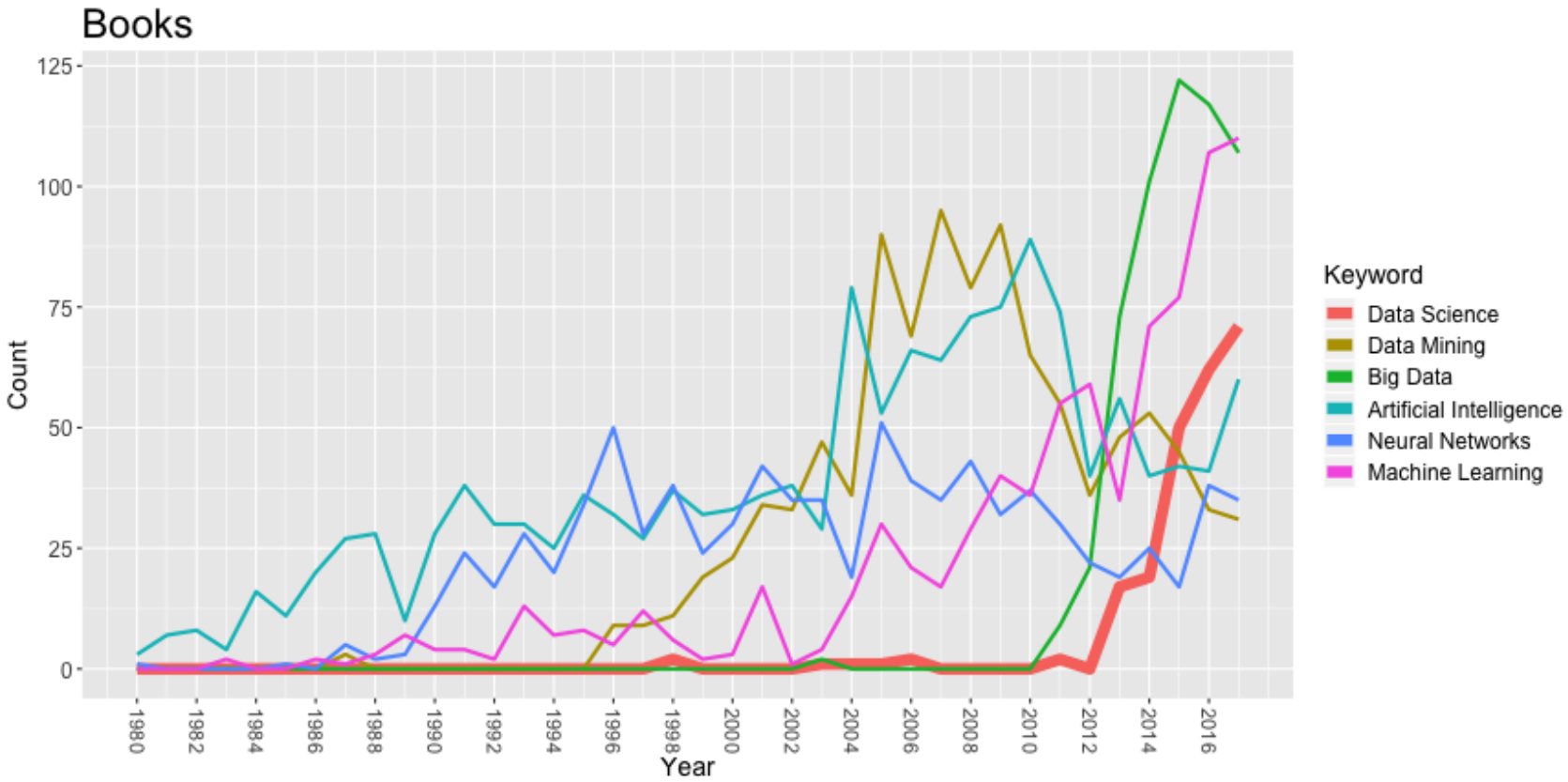


Figura : Disponível em: http://www.lac.inpe.br/ rafael.santos/r.html.

Data Science

Resumindo...

Data Science

Resumindo...

- Campo multidisciplinar

Data Science

Resumindo...

- Campo multidisciplinar
- Extrair conhecimento

Data Science

Resumindo...

- Campo multidisciplinar
- Extrair conhecimento
- Tomada de decisão

Data Science

Resumindo...

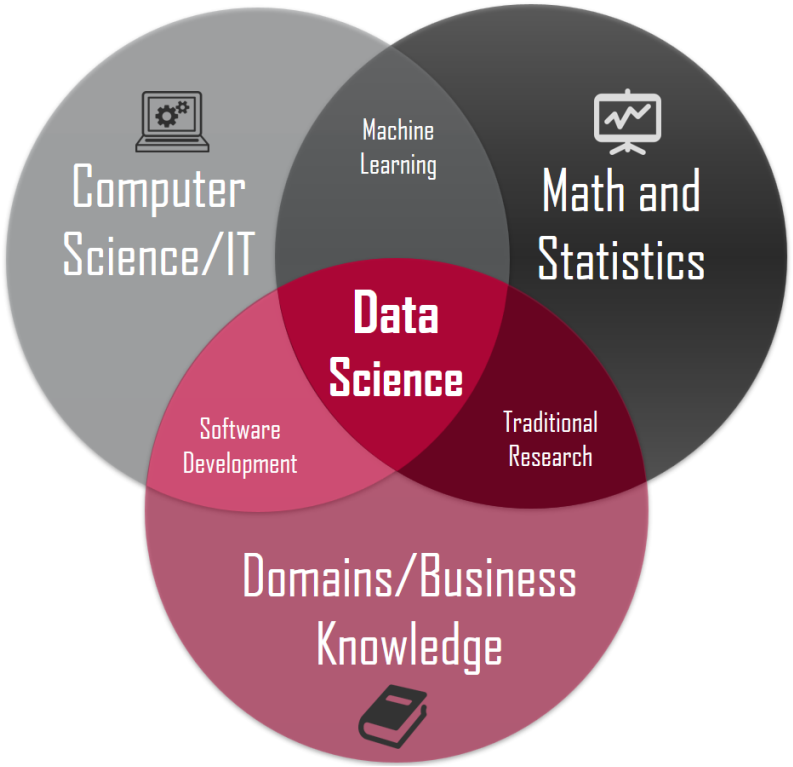


Figura : Disponível em:http://canworksmart.com/leading-a-data-driven-organization/.

Data Science

O que faz um Data Scientist?

Figura : Disponível em: https://projetodraft.com.

Data Science

O que é Data Scientist?

Data Science

O que é Data Scientist?

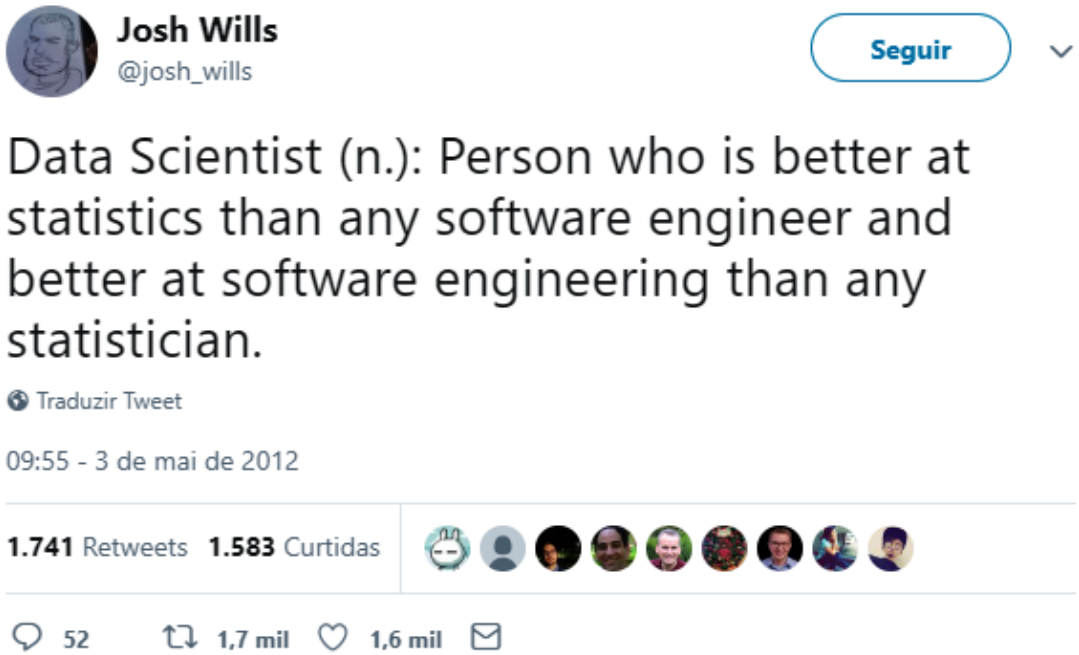


Figura : Disponível em: https://twitter.com/josh_wills.

Data Science

O que faz um Data Scientist?

Data Science

O que faz um Data Scientist?

(a) Acham e interpretam ricas fontes de dados;

Data Science

O que faz um Data Scientist?

(a) Acham e interpretam ricas fontes de dados;

(b) Gerenciam grande quantidade de dados;

Data Science

O que faz um Data Scientist?

(a) Acham e interpretam ricas fontes de dados;

(b) Gerenciam grande quantidade de dados;

(c) Criam visualizações para entender os dados;

Data Science

O que faz um Data Scientist?

(a) Acham e interpretam ricas fontes de dados;

(b) Gerenciam grande quantidade de dados;

(c) Criam visualizações para entender os dados;

(d) Criam modelos matemáticos ou estatísticos;

Data Science

O que faz um Data Scientist?

(a) Acham e interpretam ricas fontes de dados;

(b) Gerenciam grande quantidade de dados;

(c) Criam visualizações para entender os dados;

(d) Criam modelos matemáticos ou estatísticos;

(e) Apresentam e comunicam as descobertas encontradas nos dados.

Manipulação de dados

Manipulação de dados

(a) Dados estruturados;

Manipulação de dados

(a) Dados estruturados;

(b) Dados não estruturados (exemplos:e-mails, twitters, PDF, imagens)

Manipulação de dados

(a) Dados estruturados;

(b) Dados não estruturados (exemplos:e-mails, twitters, PDF, imagens)

(c) Dados semiestruturados (exemplos: JSON, XML, HTML, YAML).

Manipulação de dados

Na prática, os dados nunca estarão do jeito que queremos!

Manipulação de dados

Na prática, os dados nunca estarão do jeito que queremos!

(a) Transformar;

Manipulação de dados

Na prática, os dados nunca estarão do jeito que queremos!

(a) Transformar;

(b) Reestruturar (dados incompletos, incosistentes, redundantes);

Manipulação de dados

Na prática, os dados nunca estarão do jeito que queremos!

(a) Transformar;

(b) Reestruturar (dados incompletos, incosistentes, redundantes);

(c) Limpar;

(d) Amostragem - volume de dados muito alto (*big data*);

Manipulação de dados

Na prática, os dados nunca estarão do jeito que queremos!

(a) Transformar;

(b) Reestruturar (dados incompletos, incosistentes, redundantes);

(c) Limpar;

(d) Amostragem - volume de dados muito alto (*big data*);

(e) Agregar e juntar.

Manipulação de dados

Na prática, os dados nunca estarão do jeito que queremos!

(a) Transformar;

(b) Reestruturar (dados incompletos, incosistentes, redundantes);

(c) Limpar;

(d) Amostragem - volume de dados muito alto (*big data*);

(e) Agregar e juntar.

80 % of data analysis is spent on the process of cleaning and preparing the data (Dasu and Johnson, 2003).

Software R

Por que usar o R?

Software R

Por que usar o R?

(a) Gratuito

Software R

Por que usar o R?

(a) Gratuito

(b) Tradução da matemática para o cógido é direta

Software R

Por que usar o R?

(a) Gratuito

(b) Tradução da matemática para o cógido é direta

(c) É uma linguagem de programação

Software R

Por que usar o R?

(a) Gratuito

(b) Tradução da matemática para o cógido é direta

(c) É uma linguagem de programação

(d) Atualizado constantemente

Software R

Por que usar o R?

(a) Gratuito

(b) Tradução da matemática para o cógido é direta

(c) É uma linguagem de programação

(d) Atualizado constantemente

(e) Enorme número de pacotes

Software R

Por que usar o R?

(a) Gratuito

(b) Tradução da matemática para o cógido é direta

(c) É uma linguagem de programação

(d) Atualizado constantemente

(e) Enorme número de pacotes

(f) Comunidade ativa e crescente.

Software R

Por que usar o R?

Figura : Disponível em: https://www.r-project.org/.

Software R

Por que usar o R?

Figura : Disponível em: https://www.r-project.org/.

Figura : Disponível em: https://www.rstudio.com/.

Software R

Alguns pacotes importantes para manipulação de dados:

(a) tidyr

Software R

Alguns pacotes importantes para manipulação de dados:

(a) tidyr

(b) purrr

Software R

Alguns pacotes importantes para manipulação de dados:

(a) tidyr

(b) purrr

(c) dplyr

Software R

Alguns pacotes importantes para manipulação de dados:

(a) tidyr

(b) purrr

(c) dplyr

(d) ggplot2 (veja também - plotly)

Software R

Alguns pacotes importantes para manipulação de dados:

(a) tidyr
(b) purrr
(c) dplyr
(d) ggplot2 (veja também - plotly)
(e) tidyverse (tidyr + purrr + dplyr + ggplot2 + stringr)

Software R

Alguns pacotes importantes para manipulação de dados:

Software R

Alguns pacotes importantes para manipulação de dados:

Figura : http://qualimetria.ufsc.br/files/2018/03/Projeto-T%C3%ADpico-de-Data-Science-com-R.pdf.

Software R

Alguns pacotes importantes para Machine Learning:

Software R

Alguns pacotes importantes para Machine Learning:

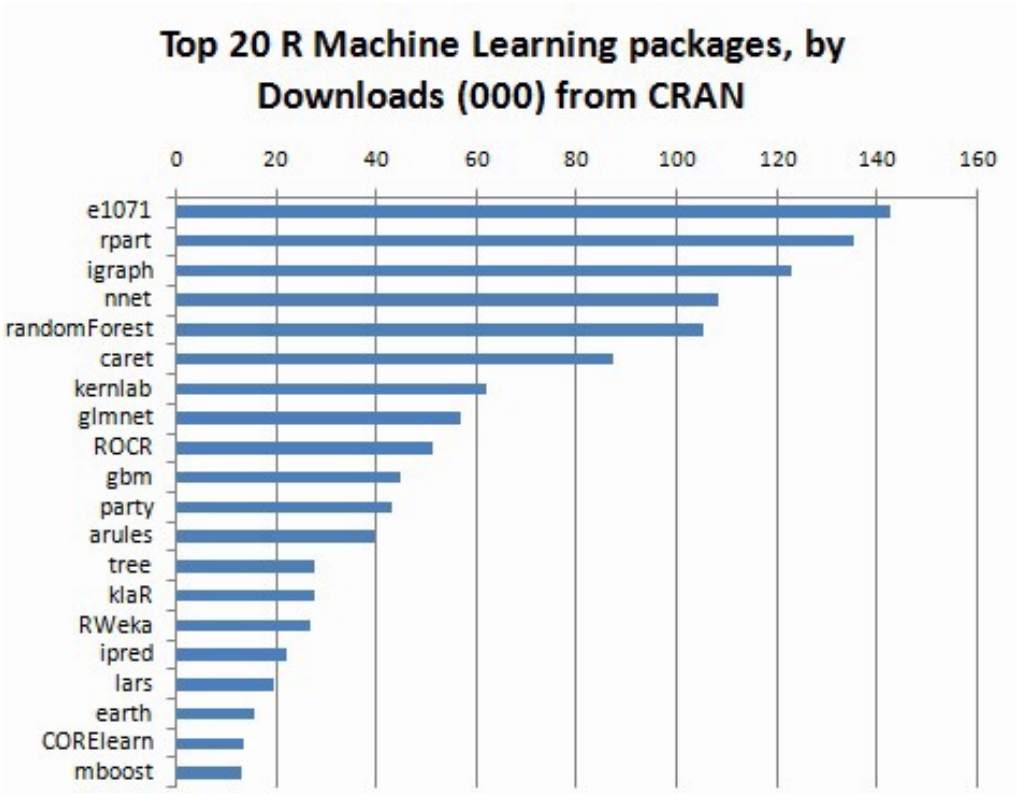


Figura : https://www.kdnuggets.com/2015/06/top-20-r-machine-learning-packages.html.

Software Python

Por que usar o Python?

Software Python

Por que usar o Python?

- Linguagem simples e fácil de aprender

Software Python

Por que usar o Python?

- Linguagem simples e fácil de aprender
- Linguagem de programação

Software Python

Por que usar o Python?

- Linguagem simples e fácil de aprender
- Linguagem de programação
- Código fonte aberto e multiplataforma

Software Python

Por que usar o Python?

- Linguagem simples e fácil de aprender
- Linguagem de programação
- Código fonte aberto e multiplataforma
- Gratuito

Software Python

Por que usar o Python?

- Linguagem simples e fácil de aprender
- Linguagem de programação
- Código fonte aberto e multiplataforma
- Gratuito
- Enorme número de pacotes

Software Python

Por que usar o Python?

- Linguagem simples e fácil de aprender
- Linguagem de programação
- Código fonte aberto e multiplataforma
- Gratuito
- Enorme número de pacotes
- Grande comunidade de usuários.

Software Python

Figura : https://www.python.org/downloads/.

Software Python

Figura : https://www.python.org/downloads/.

Figura : https://www.anaconda.com/distribution/.

Software Python

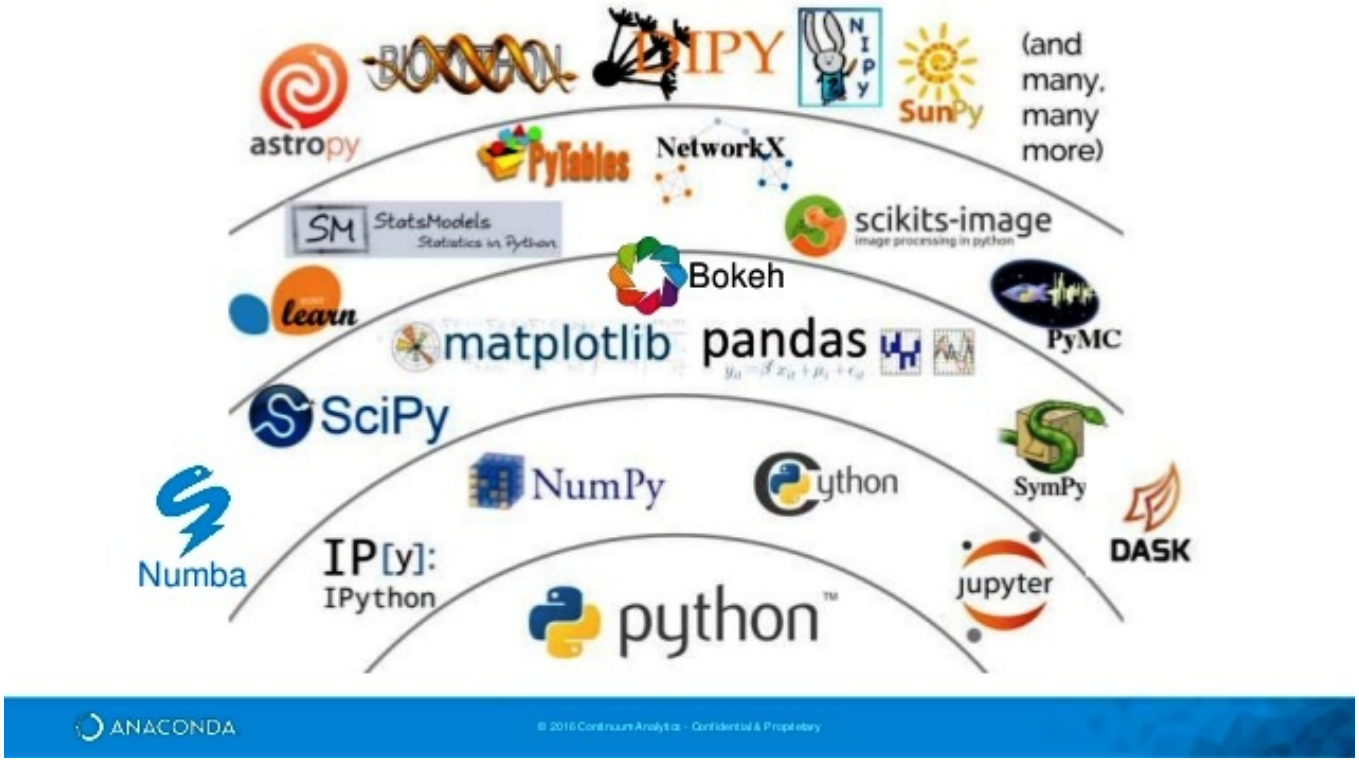


Figura : https://www.slideshare.net/teoliphant/python-for-data-science-with-anaconda.

Modelagem dos dados

Modelagem dos dados

(a) Classification: Decidir se algo pertence a uma certa categoria

Modelagem dos dados

(a) Classification: Decidir se algo pertence a uma certa categoria

(b) Scoring ou Regression: Predizer ou estimar um valor numérico

Modelagem dos dados

(a) Classification: Decidir se algo pertence a uma certa categoria

(b) Scoring ou Regression: Predizer ou estimar um valor numérico

(c) Ranking: Aprender a ordenar itens por preferências

Modelagem dos dados

(a) Classification: Decidir se algo pertence a uma certa categoria
(b) Scoring ou Regression: Predizer ou estimar um valor numérico
(c) Ranking: Aprender a ordenar itens por preferências
(d) Clustering: Agrupar itens por similaridade.

Modelagem dos dados

(a) Classification: Decidir se algo pertence a uma certa categoria

(b) Scoring ou Regression: Predizer ou estimar um valor numérico

(c) Ranking: Aprender a ordenar itens por preferências

(d) Clustering: Agrupar itens por similaridade.

(e) Finding relations: Achar correlações ou causas potenciais de efeitos observados.

Machine Learning

Uma definição

Aprendizado de Máquina é uma área de Inteligência Artificial(IA) cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática.

Machine Learning

Uma definição

Aprendizado de Máquina é uma área de Inteligência Artificial(IA) cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática.

Mitchell (1997) forneceu uma definição mais formal amplamente citada: "Diz-se que um programa de computador aprende pela experiência E, com respeito a algum tipo de tarefa T e performance P, se sua performance P nas tarefas em T, na forma medida por P, melhoram com a experiência E".

ML - Motivação

Existem inúmeras aplicações de Aprendizado de Máquinas, podemos citar:

(a) Análise de risco de crédito em bancos

ML - Motivação

Existem inúmeras aplicações de Aprendizado de Máquinas, podemos citar:

(a) Análise de risco de crédito em bancos

(b) Reconhecimento de palavras faladas

ML - Motivação

Existem inúmeras aplicações de Aprendizado de Máquinas, podemos citar:

(a) Análise de risco de crédito em bancos

(b) Reconhecimento de palavras faladas

(c) Detecção do uso fraudulento de cartões de crédito

ML - Motivação

Existem inúmeras aplicações de Aprendizado de Máquinas, podemos citar:

(a) Análise de risco de crédito em bancos

(b) Reconhecimento de palavras faladas

(c) Detecção do uso fraudulento de cartões de crédito

(d) Filtragem de spam nos e-mails

ML - Motivação

Existem inúmeras aplicações de Aprendizado de Máquinas, podemos citar:

(a) Análise de risco de crédito em bancos

(b) Reconhecimento de palavras faladas

(c) Detecção do uso fraudulento de cartões de crédito

(d) Filtragem de spam nos e-mails

(e) Novos modelos de precificação de ativos

ML - Motivação

Existem inúmeras aplicações de Aprendizado de Máquinas, podemos citar:

(a) Análise de risco de crédito em bancos

(b) Reconhecimento de palavras faladas

(c) Detecção do uso fraudulento de cartões de crédito

(d) Filtragem de spam nos e-mails

(e) Novos modelos de precificação de ativos

(f) Análise de sentimento

ML - Motivação

Existem inúmeras aplicações de Aprendizado de Máquinas, podemos citar:

(a) Análise de risco de crédito em bancos
(b) Reconhecimento de palavras faladas
(c) Detecção do uso fraudulento de cartões de crédito
(d) Filtragem de spam nos e-mails
(e) Novos modelos de precificação de ativos
(f) Análise de sentimento
(g) Recomendação de conteúdo

ML - Motivação

Existem inúmeras aplicações de Aprendizado de Máquinas, podemos citar:

(a) Análise de risco de crédito em bancos
(b) Reconhecimento de palavras faladas
(c) Detecção do uso fraudulento de cartões de crédito
(d) Filtragem de spam nos e-mails
(e) Novos modelos de precificação de ativos
(f) Análise de sentimento
(g) Recomendação de conteúdo
(h) etc...

Tipos de Aprendizagem

No aprendizado supervisionado é fornecido ao algoritmo de aprendizado, ou indutor, um conjunto de exemplos de treinamento para os quais o rótulo da classe associada conhecido.

Tipos de Aprendizagem

No aprendizado supervisionado é fornecido ao algoritmo de aprendizado, ou indutor, um conjunto de exemplos de treinamento para os quais o rótulo da classe associada conhecido.

Aprendizagem Supervisionada

Dado um conjunto de observações ou exemplos rotulados, isto é, conjunto de observações em que a classe, denominada também atributo meta, de cada exemplo é conhecida, o objetivo é encontrar uma hipótese capaz de classificar novas observações entre as classes já existentes.

Tipos de Aprendizagem

No aprendizado supervisionado é fornecido ao algoritmo de aprendizado, ou indutor, um conjunto de exemplos de treinamento para os quais o rótulo da classe associada conhecido.

Aprendizagem Supervisionada

Dado um conjunto de observações ou exemplos rotulados, isto é, conjunto de observações em que a classe, denominada também atributo meta, de cada exemplo é conhecida, o objetivo é encontrar uma hipótese capaz de classificar novas observações entre as classes já existentes.
Fornecemos a 'resposta correta' durante o treinamento, isto é, as classes são conhecidas a priori.

Aprendizagem Supervisionada

Dado um conjunto de dados, este pode ser dividido de duas formas:

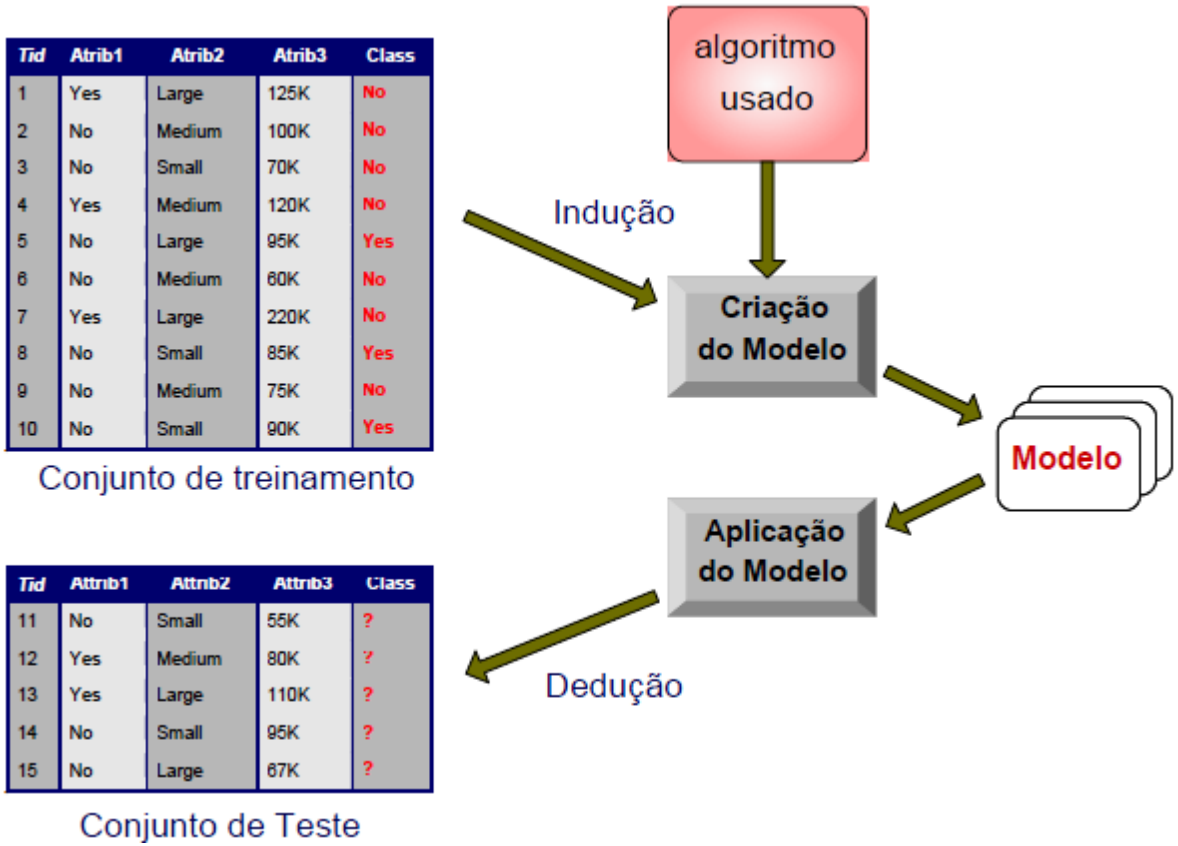


Figura : Treinamento e Teste.

Aprendizagem Supervisionada

No aprendizado supervisionado os exemplos são rotulados. Os exemplos são ditos supervisionados pois, além de conter a entrada (vetor de atributos), possui também o rótulo ou saída (classe).

Aprendizagem Supervisionada

No aprendizado supervisionado os exemplos são rotulados. Os exemplos são ditos supervisionados pois, além de conter a entrada (vetor de atributos), possui também o rótulo ou saída (classe).

Assim, o objetivo do algoritmo de indução é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados, isto é, exemplos que não tenham o rótulo da classe.

Aprendizagem Supervisionada

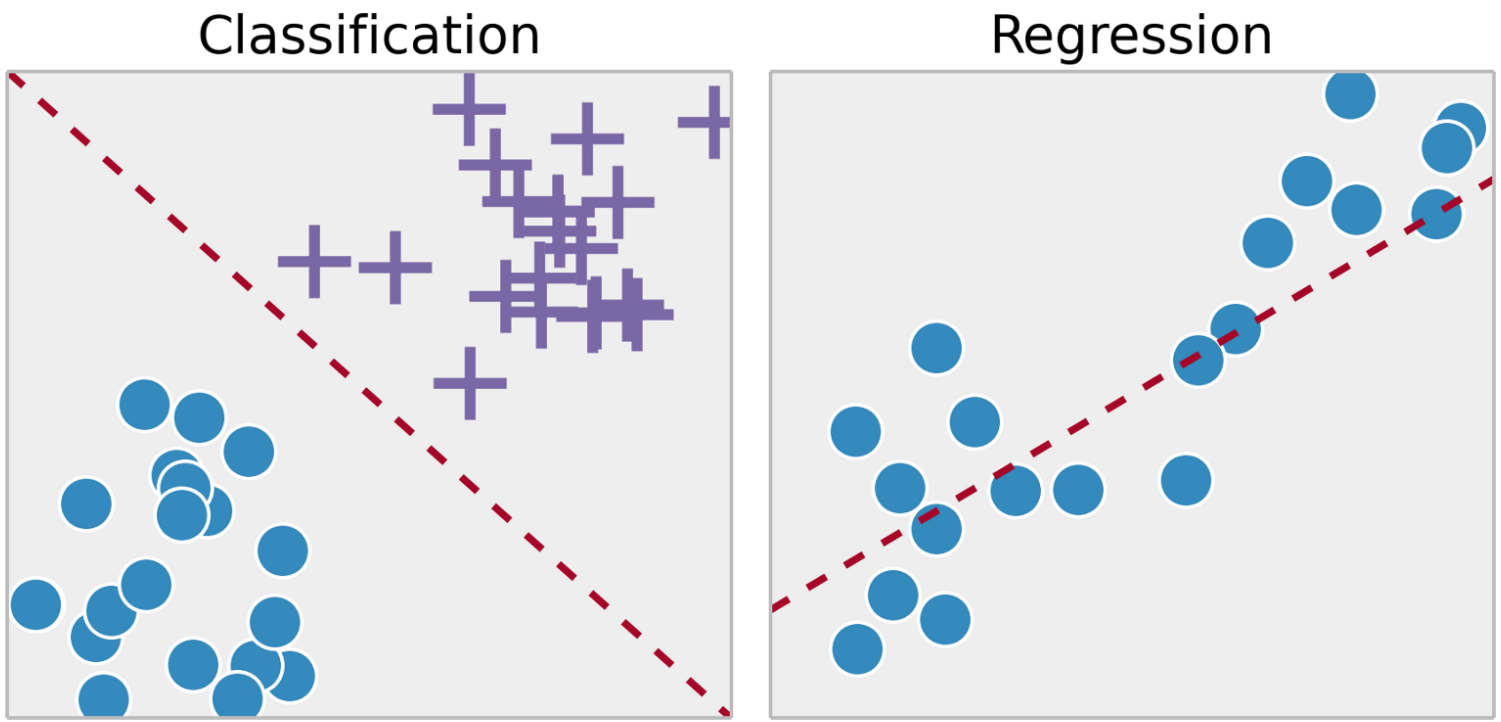


Figura : Classificação e Regressão.

Aprendizagem Supervisionada

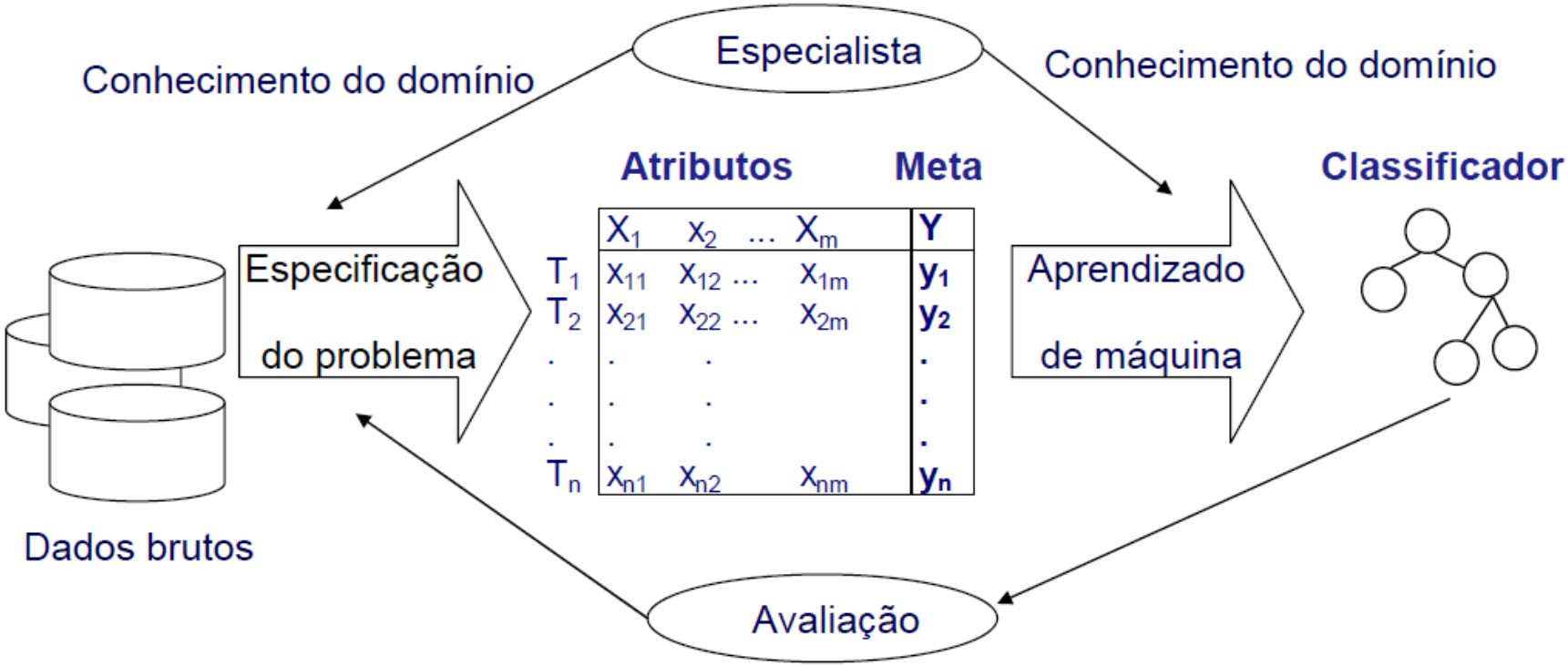
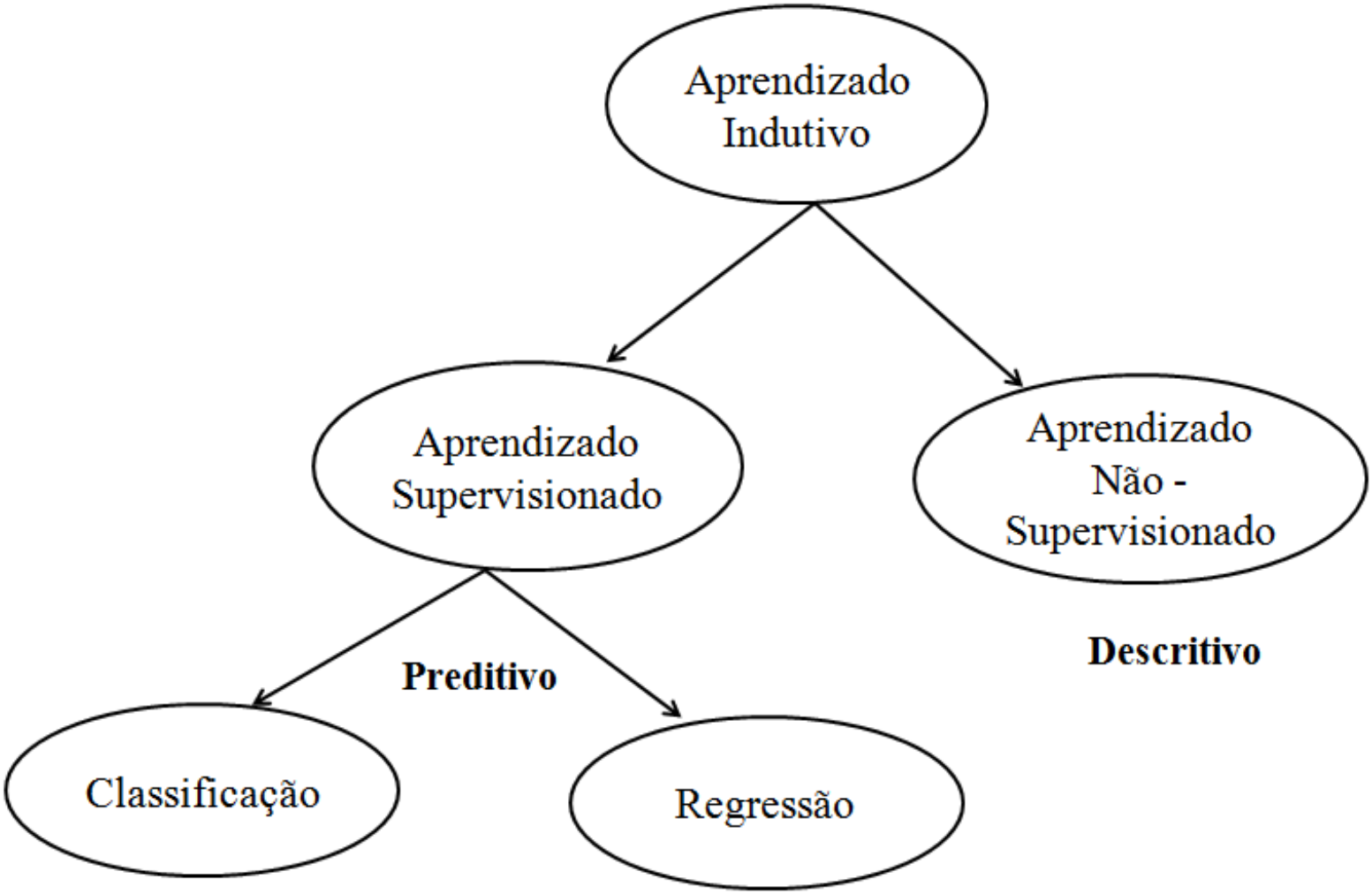


Figura : Processo de Classificação.

Hierarquia do Aprendizado



Introdução aos Modelos Preditivos

(a) Árvores de Decisão

Introdução aos Modelos Preditivos

(a) Árvores de Decisão

(b) *Random Forest*

Introdução aos Modelos Preditivos

(a) Árvores de Decisão

(b) *Random Forest*

(c) K-vizinhos mais próximos (*k*-NN)

Introdução aos Modelos Preditivos

(a) Árvores de Decisão
(b) *Random Forest*
(c) K-vizinhos mais próximos (*k*-NN)
(d) Naive Bayes

Introdução aos Modelos Preditivos

(a) Árvores de Decisão
(b) *Random Forest*
(c) K-vizinhos mais próximos (*k*-NN)
(d) Naive Bayes
(e) *Support Vector Machines (SVMs)*

Árvores de Decisão

(a) Um dos métodos de aprendizagem mais conhecidos e utilizados para fazer classificação;

Árvores de Decisão

(a) Um dos métodos de aprendizagem mais conhecidos e utilizados para fazer classificação;

(b) É essencialmente uma série de declarações *if-elses*, que quando aplicados a um registro de uma base de dados resultam na classificação daquele registro;

Árvores de Decisão

(a) Um dos métodos de aprendizagem mais conhecidos e utilizados para fazer classificação;

(b) É essencialmente uma série de declarações *if-elses*, que quando aplicados a um registro de uma base de dados resultam na classificação daquele registro;

(c) Não exige a normalização dos dados;

Árvores de Decisão

(a) Um dos métodos de aprendizagem mais conhecidos e utilizados para fazer classificação;

(b) É essencialmente uma série de declarações *if-elses*, que quando aplicados a um registro de uma base de dados resultam na classificação daquele registro;

(c) Não exige a normalização dos dados;

(d) Pode receber tanto dados numéricos quanto categóricos (não assumem nenhuma distribuição para os dados);

Árvores de Decisão

(a) Um dos métodos de aprendizagem mais conhecidos e utilizados para fazer classificação;

(b) É essencialmente uma série de declarações *if-elses*, que quando aplicados a um registro de uma base de dados resultam na classificação daquele registro;

(c) Não exige a normalização dos dados;

(d) Pode receber tanto dados numéricos quanto categóricos (não assumem nenhuma distribuição para os dados);

(e) Não é influenciado por *outliers* e valores faltantes.

Árvores de Decisão

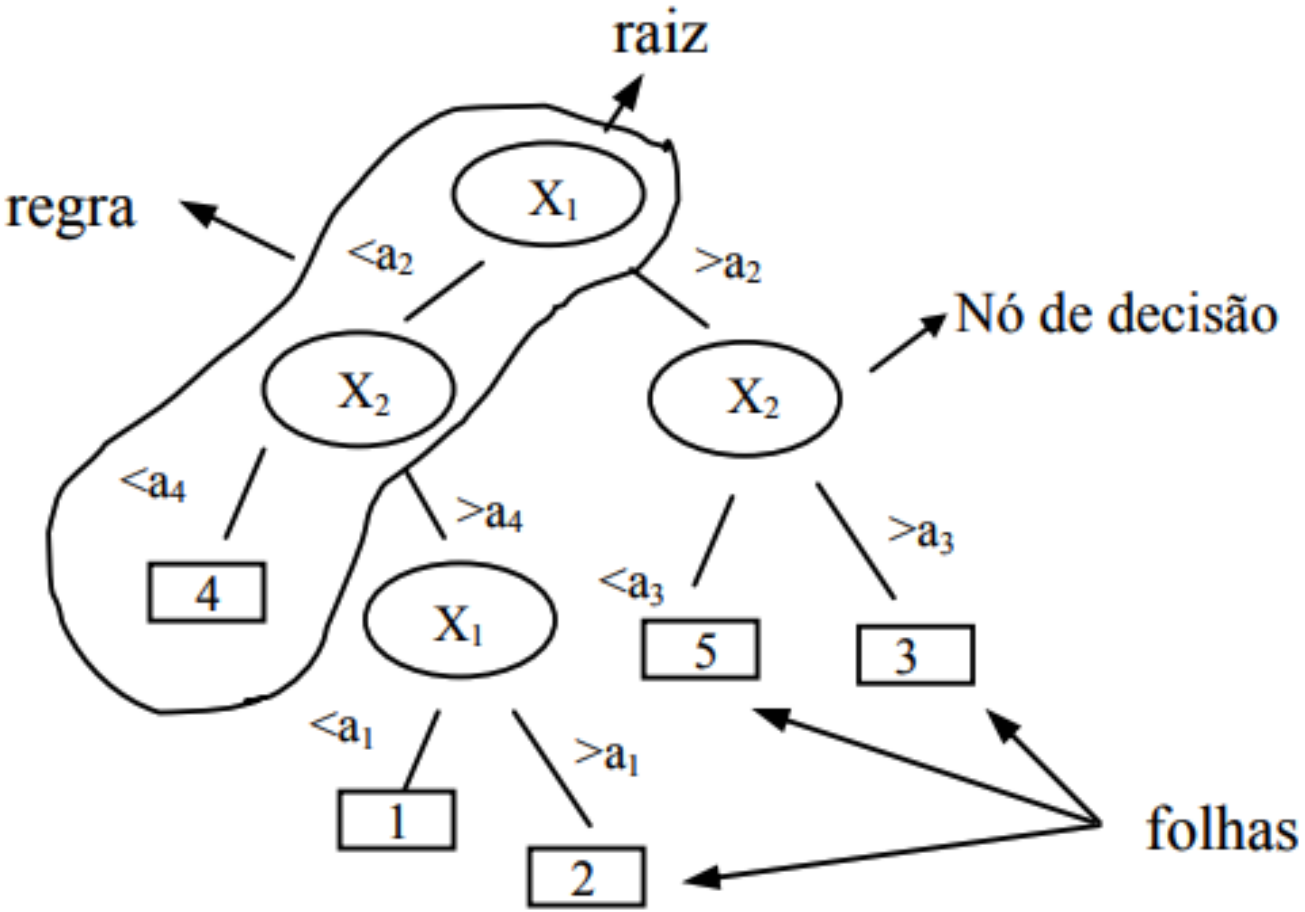


Figura : Representação de uma árvore de decisão - Fonte: Gama, 2004

Árvores de Decisão

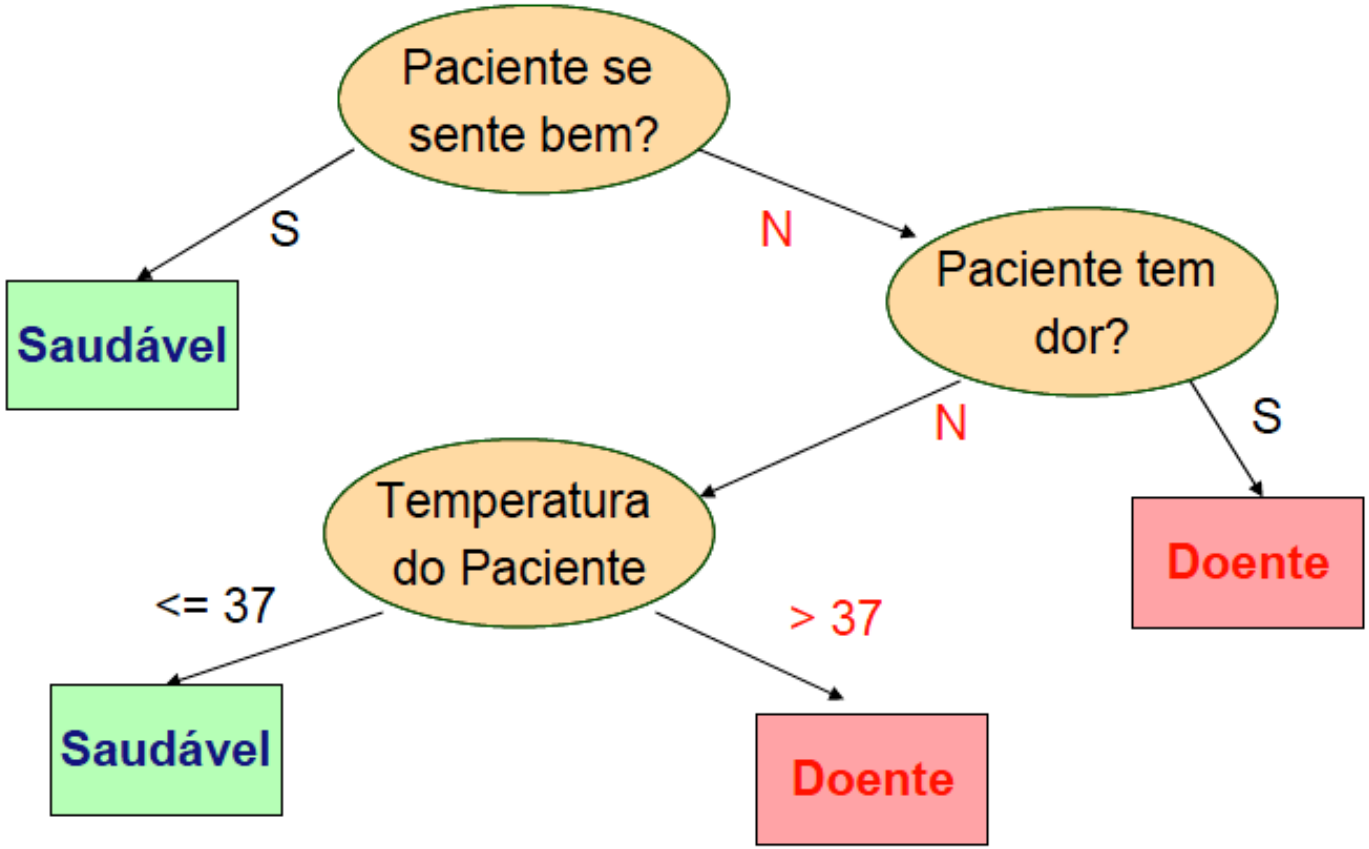


Figura : Exemplo de uma árvore de decisão.

Árvores de Decisão

Para construirmos uma árvore de decisão precisamos decidir quais perguntas fazer e em qual ordem. Para tanto, podemos utilizar o conceito de Entropia.

Árvores de Decisão

Para construirmos uma árvore de decisão precisamos decidir quais perguntas fazer e em qual ordem. Para tanto, podemos utilizar o conceito de Entropia.
Em termos matemáticos, se p_i é a proporção de dados definidos como classes c_i,podemos definir a entropia como:

$$H(S) = -p_1 log_2 p_1 - ... - p_n log_2 p_n. \quad (1)$$

Árvores de Decisão

Ao dividir um conjunto de dados S em subconjuntos $S_1, ..., S_m$ contendo porções de dados $q_1, ..., q_m$, então podemos computar a entropia da partição como uma soma ponderada:

Árvores de Decisão

Ao dividir um conjunto de dados S em subconjuntos $S_1, ..., S_m$ contendo porções de dados $q_1, ..., q_m$, então podemos computar a entropia da partição como uma soma ponderada:

$$H = q_1 H(S_1) + ... + q_m H(S_m). \quad (2)$$

Outro conceito muito relacionado à entropia é o ganho informacional. O ganho de informação ocorre quando uma nova subdivisão dos dados provoca uma redução na entropia.

Árvores de Decisão

O ganho informacional é baseado na redução da entropia depois de uma divisão do conjunto de dados baseado em determinada regra. Construir uma árvore de decisão se trata de encontrar regras sobre as variáveis do modelo (ou pontos de corte) que retornam o maior ganho de informação, isto é, que tornam os ramos da árvore mais homogêneos, com menor entropia (algoritmo ID3).

Árvores de Decisão

Passos do algoritmo ID3:

Árvores de Decisão

Passos do algoritmo ID3:

1. Começar com todos os exemplos de treino;

Árvores de Decisão

Passos do algoritmo ID3:

1. Começar com todos os exemplos de treino;
2. Escolher o teste (atributo) que melhor divide os exemplos, ou seja, agrupar exemplos da mesma classe ou exemplos semelhantes;

Árvores de Decisão

Passos do algoritmo ID3:

1. Começar com todos os exemplos de treino;
2. Escolher o teste (atributo) que melhor divide os exemplos, ou seja, agrupar exemplos da mesma classe ou exemplos semelhantes;
3. Para o atributo escolhido, criar um nó filho para cada valor possível do atributo;

Árvores de Decisão

Passos do algoritmo ID3:

1. Começar com todos os exemplos de treino;
2. Escolher o teste (atributo) que melhor divide os exemplos, ou seja, agrupar exemplos da mesma classe ou exemplos semelhantes;
3. Para o atributo escolhido, criar um nó filho para cada valor possível do atributo;
4. Transportar os exemplos para cada filho tendo em conta o valor do filho;

Árvores de Decisão

Passos do algoritmo ID3:

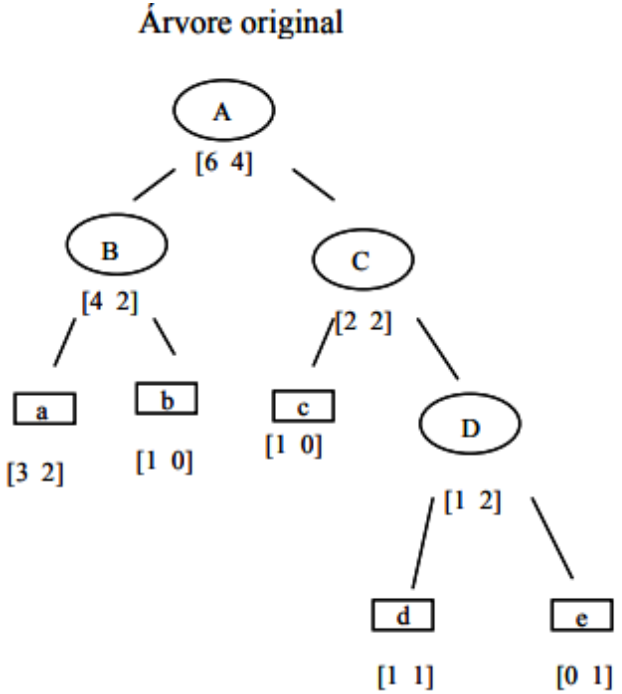
1. Começar com todos os exemplos de treino;
2. Escolher o teste (atributo) que melhor divide os exemplos, ou seja, agrupar exemplos da mesma classe ou exemplos semelhantes;
3. Para o atributo escolhido, criar um nó filho para cada valor possível do atributo;
4. Transportar os exemplos para cada filho tendo em conta o valor do filho;
5. Repetir o procedimento para cada filho não "puro". Um filho é puro quando cada atributo X tem o mesmo valor em todos os exemplos.

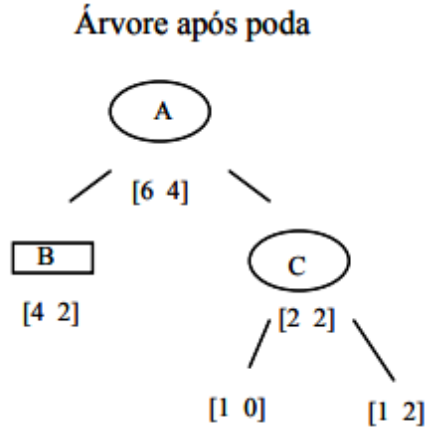
Árvores de Decisão - Poda

A podagem pode ser dada de duas circunstâncias. Pode ser usada para parar o crescimento da árvore mais cedo, chamada de pré-podagem ou poda descendente ou ainda pode acontecer com a árvore já completa, chamada de pós-podagem ou poda ascendente.

Árvores de Decisão - Poda

A podagem pode ser dada de duas circunstâncias. Pode ser usada para parar o crescimento da árvore mais cedo, chamada de pré-podagem ou poda descendente ou ainda pode acontecer com a árvore já completa, chamada de pós-podagem ou poda ascendente.





Random Forest

Random Forest é um algoritmo de aprendizagem de máquina flexível e fácil de usar que produz excelentes resultados na maioria das vezes, mesmo sem ajuste de hiperparâmetros.

Random Forest

Random Forest é um algoritmo de aprendizagem de máquina flexível e fácil de usar que produz excelentes resultados na maioria das vezes, mesmo sem ajuste de hiperparâmetros.
O algoritmo de *Random Forest* cria várias árvores de decisão e as combina para obter uma predição com maior acurácia e mais estável.

Random Forest

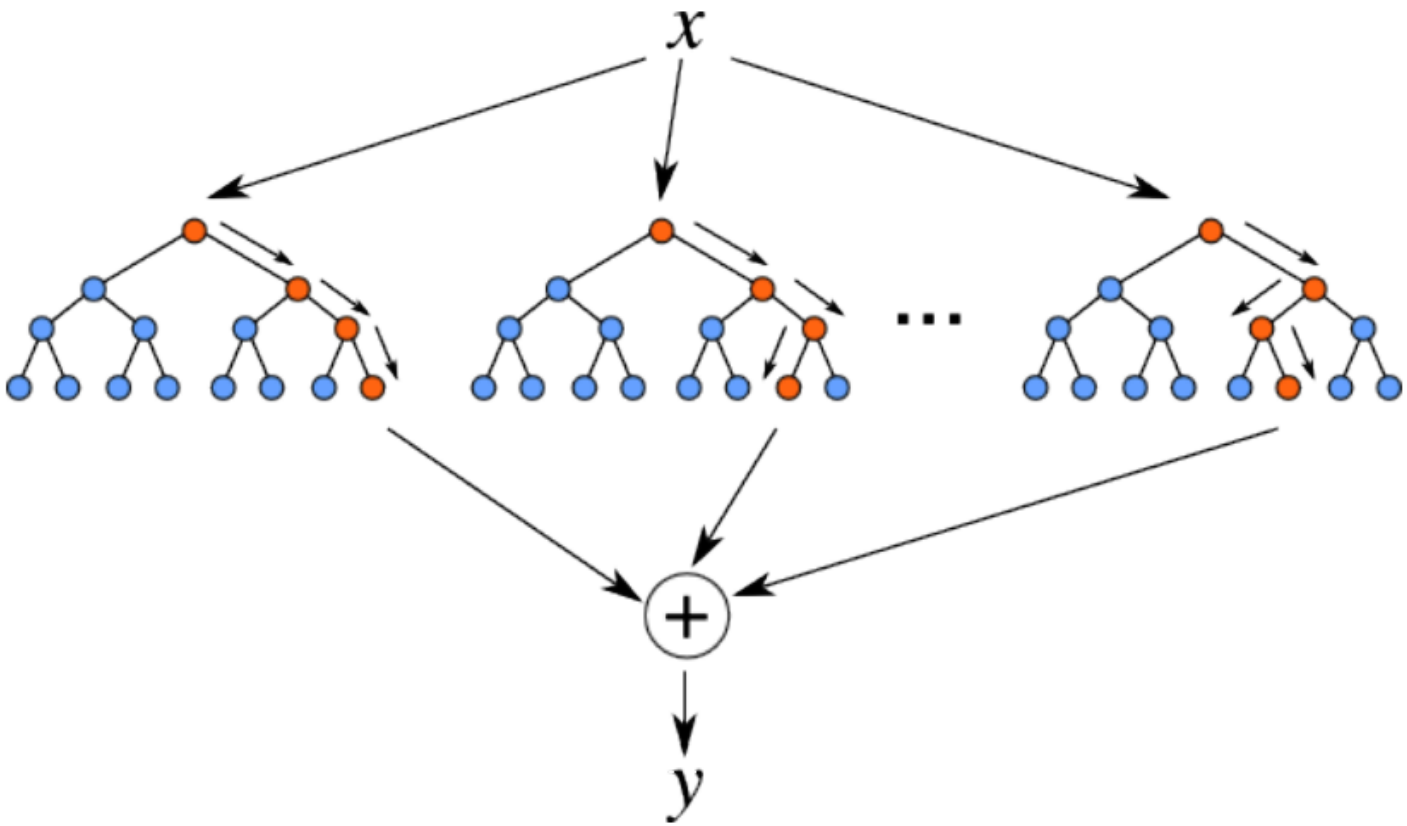


Figura : Ilustração da lógica por trás do algoritmo Random Forest.

Aprendizado Baseado em Instâncias - *k*-NN

A hipótese básica aqui é que dados similares tendem a estar concentrados em uma mesma região no espaço de entrada. De forma análoga, dados que não são similares estarão distantes entre si.

Aprendizado Baseado em Instâncias - *k*-NN

A hipótese básica aqui é que dados similares tendem a estar concentrados em uma mesma região no espaço de entrada. De forma análoga, dados que não são similares estarão distantes entre si.

Dentre os algoritmos utilizados no aprendizado baseado em instâncias destaca-se o *k* - NN (k-*nearest neighbors* - vizinho mais próximo).

Aprendizado Baseado em Instâncias - *k*-NN

O algoritmo *k*-NN é um algoritmo de aprendizado superivisionado baseado em instâncias (*instance based learning*) do tipo *lazy* (porque armazena os dados de treino e realiza uma única etapa para fazer a classificação).

Aprendizado Baseado em Instâncias - *k*-NN

O algoritmo *k*-NN é um algoritmo de aprendizado superivisionado baseado em instâncias (*instance based learning*) do tipo *lazy* (porque armazena os dados de treino e realiza uma única etapa para fazer a classificação).

(a) O aprendizado baseado em instâncias está fundamentado na aplicação direta do conceito de similaridade, que pode ser entendido como o quão próximas duas instâncias estão.

Aprendizado Baseado em Instâncias - *k*-NN

O algoritmo *k*-NN é um algoritmo de aprendizado superivisionado baseado em instâncias (*instance based learning*) do tipo *lazy* (porque armazena os dados de treino e realiza uma única etapa para fazer a classificação).

(a) O aprendizado baseado em instâncias está fundamentado na aplicação direta do conceito de similaridade, que pode ser entendido como o quão próximas duas instâncias estão.

(b) Um exemplar é classficado pela "votação da maioria", realizada junto aos seus vizinhos no conjunto de treinamento armazenado.

Aprendizado Baseado em Instâncias - *k*-NN

O algoritmo *k*-NN é um algoritmo de aprendizado superivisionado baseado em instâncias (*instance based learning*) do tipo *lazy* (porque armazena os dados de treino e realiza uma única etapa para fazer a classificação).

(a) O aprendizado baseado em instâncias está fundamentado na aplicação direta do conceito de similaridade, que pode ser entendido como o quão próximas duas instâncias estão.

(b) Um exemplar é classficado pela "votação da maioria", realizada junto aos seus vizinhos no conjunto de treinamento armazenado.

(c) A relação de proximidade para verificação dos vizinhos é quantificada por uma métrica de distância.

Aprendizado Baseado em Instâncias - *k*-NN

O algoritmo *k*-NN é um algoritmo de aprendizado superivisionado baseado em instâncias (*instance based learning*) do tipo *lazy* (porque armazena os dados de treino e realiza uma única etapa para fazer a classificação).

(a) O aprendizado baseado em instâncias está fundamentado na aplicação direta do conceito de similaridade, que pode ser entendido como o quão próximas duas instâncias estão.

(b) Um exemplar é classficado pela "votação da maioria", realizada junto aos seus vizinhos no conjunto de treinamento armazenado.

(c) A relação de proximidade para verificação dos vizinhos é quantificada por uma métrica de distância.

(d) Diferentes métricas podem gerar diferentes classificações.

Algoritmo *k*-NN

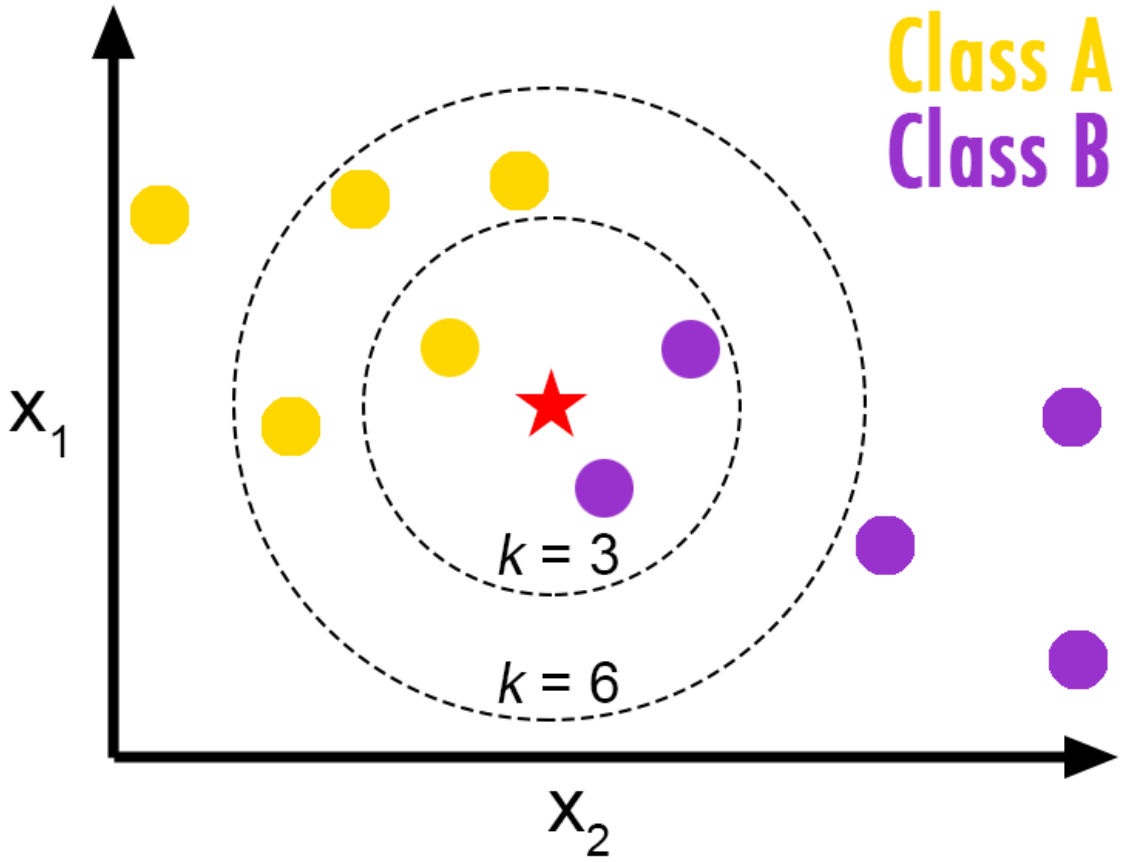


Figura : Classificação por meio de vizinhos mais próximos.

Algoritmo *k*-NN

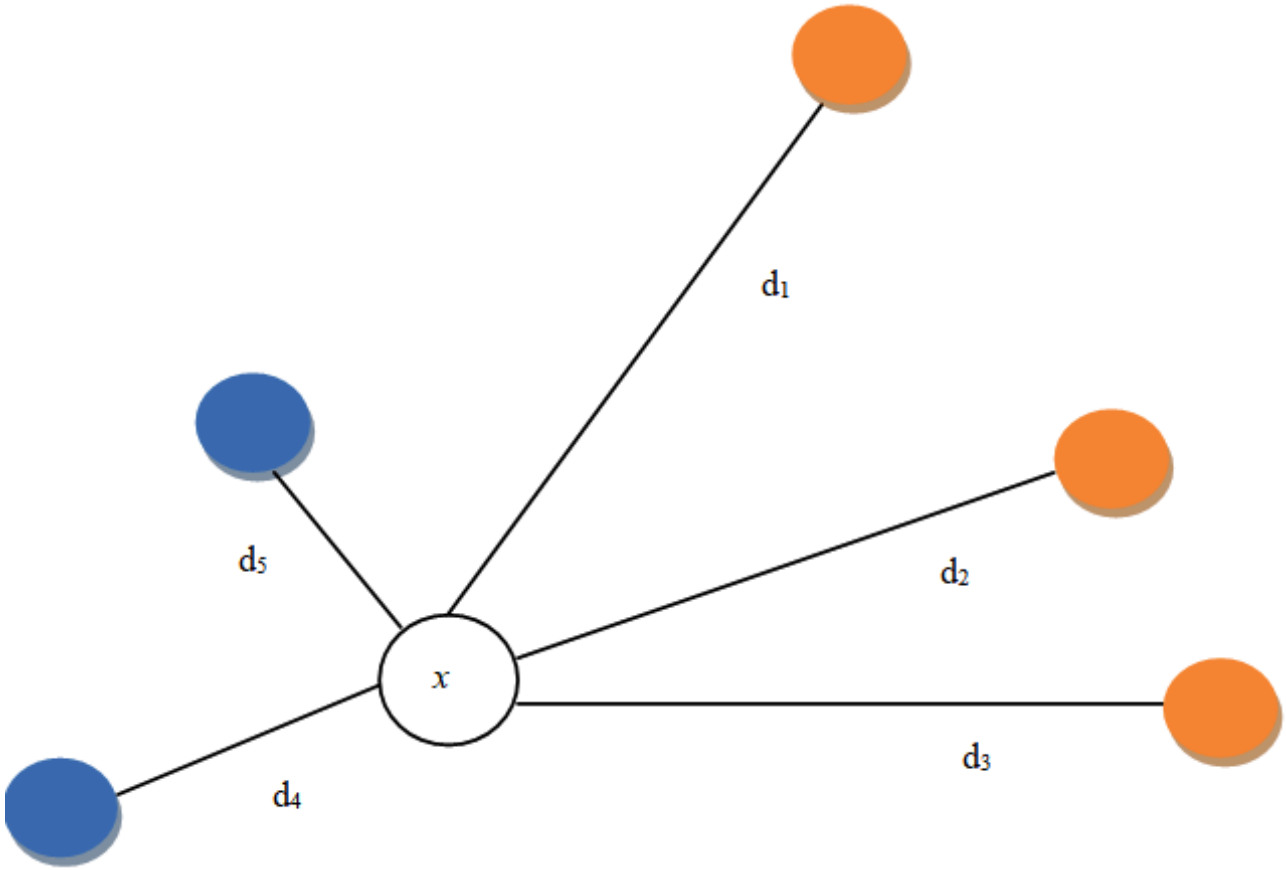


Figura : Classificação por meio de vizinhos mais próximos.

Algoritmo *k*-NN

A distância euclidiana é a métrica mais utilizada. A equação (3) encontra um valor que define a distância entre dois pontos (x e y) no espaço n - dimensional.

$$dist = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}. \quad (3)$$

O ponto y mais próximo de x será aquele com a menor distância.

Algoritmo *k*-NN

A distância euclidiana é a métrica mais utilizada. A equação (3) encontra um valor que define a distância entre dois pontos (x e y) no espaço n - dimensional.

$$dist = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}. \quad (3)$$

O ponto y mais próximo de x será aquele com a menor distância.

Antes de iniciar o algoritmo é necessário definir dois parâmetros: (1) o valor de k, que define o número de vizinhos mais próximos e (2) a medida de distância.

Naive Bayes

No processo de aprendizado por meio do algoritmo Naive Bayes o processo é da forma:

O classificador Naive Bayes aprende a partir dos dados de treinamento a probabilidade condicional de cada atributo dado o valor da classe.

Naive Bayes

No processo de aprendizado por meio do algoritmo Naive Bayes o processo é da forma:
O classificador Naive Bayes aprende a partir dos dados de treinamento a probabilidade condicional de cada atributo dado o valor da classe.

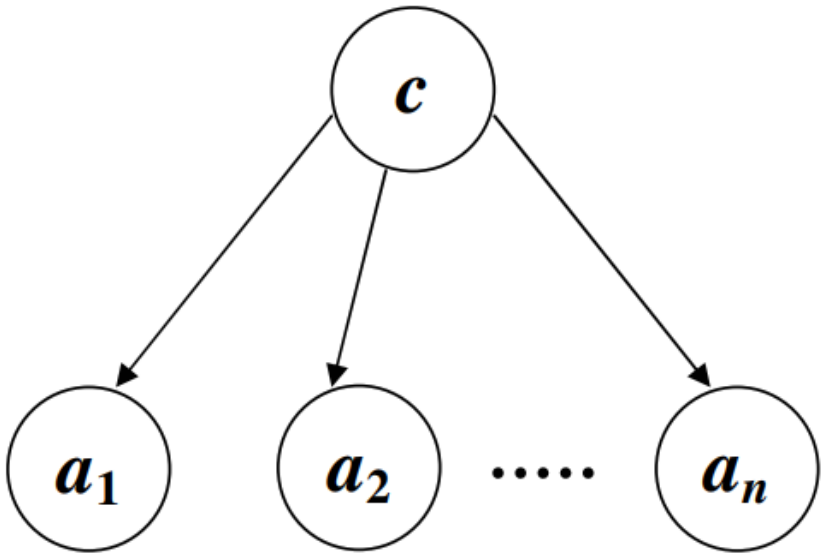


Figura : Classificador Naive Bayes (Wanke et.al.)

Naive Bayes

A classificação é feita aplicando-se o Teorema de Bayes. Dados os valores dos atributos no caso teste, e escolhendo a que resulta em maior probabilidade, isto é, queremos encontrar a classe mais provável $c \in C$, sendo fornecidos os exemplos de treinamento $(a_2, a_2, ..., a_n)$, isto é:

$$c_{teste} = arg \max_{c} \frac{P(a_1, a_2, ..., a_n|c) P(c)}{P(a_1, a_2, ..., a_n)} \quad (4)$$

Naive Bayes

A classificação é feita aplicando-se o Teorema de Bayes. Dados os valores dos atributos no caso teste, e escolhendo a que resulta em maior probabilidade, isto é, queremos encontrar a classe mais provável $c \in C$, sendo fornecidos os exemplos de treinamento $(a_2, a_2, ..., a_n)$, isto é:

$$c_{teste} = arg \max_{c} \frac{P(a_1, a_2, ..., a_n|c) P(c)}{P(a_1, a_2, ..., a_n)} \quad (4)$$

Como o denominador é independente de c tem-se:

$$c_{teste} = arg \max_{c} P(a_1, a_2, ..., a_n|c) P(c) \quad (5)$$

Naive Bayes

A classificação é feita aplicando-se o Teorema de Bayes. Dados os valores dos atributos no caso teste, e escolhendo a que resulta em maior probabilidade, isto é, queremos encontrar a classe mais provável $c \in C$, sendo fornecidos os exemplos de treinamento $(a_2, a_2, ..., a_n)$, isto é:

$$c_{teste} = arg \max_{c} \frac{P(a_1, a_2, ..., a_n|c) P(c)}{P(a_1, a_2, ..., a_n)} \quad (4)$$

Como o denominador é independente de c tem-se:

$$c_{teste} = arg \max_{c} P(a_1, a_2, ..., a_n|c) P(c) \quad (5)$$

Logo, assumindo a independência dos atributos dada a classe:

$$c_{teste} = arg\ max\ P(c) P(a_1|c) P(a_2|c) ... P(a_n|c). \quad (6)$$

Naive Bayes

Isto é,

$$c_{teste} = \arg\max_{c} P(c) \prod_{i} P(a_i|c). \tag{7}$$

Por último, o cálculo das probabilidades é feito pela simples contagem de exemplos no conjunto de treinamento:

$$P(a_i|c) = \frac{P(a_i \cap c)}{P(c)} \tag{8}$$

Sendo,

$$P(c) = \frac{|S_c|}{|S|}. \tag{9}$$

Naive Bayes

E,

$$P\left(a_i \cap c\right) = \frac{|S_{a_i \cap c}|}{S} \tag{10}$$

Então,

$$P\left(a_i|c\right) = \frac{P\left(a_i \cap c\right)}{P(c)} = \frac{|S_{a_i \cap c}|}{|S|}\frac{|S|}{|S_c|} = \frac{|S_{a_i \cap c}|}{|S_c|}. \tag{11}$$

Naive Bayes

Vantagens:

Quando a suposição de independência é válida, um classificador Naive Bayes se apresenta melhor comparado a outros modelos, como a regressão logística e você precisa de menos dados de treinamento.

Naive Bayes

Vantagens:

Quando a suposição de independência é válida, um classificador Naive Bayes se apresenta melhor comparado a outros modelos, como a regressão logística e você precisa de menos dados de treinamento.
É fácil e rápido prever a classe de conjuntos de dados de teste. Ele funciona bem tanto para variáveis categóricas como para variáveis numéricas (para variáveis numéricas, é assumida a distribuição normal).

Naive Bayes

Um problema:

Se a variável categórica tiver uma categoria (no conjunto de dados de teste), que não foi observada no conjunto de dados de treinamento, então o modelo atribuirá uma probabilidade de 0 (zero) e será incapaz de fazer uma previsão. Isso geralmente é conhecido como "Frequência zero". Para resolver isso, podemos usar a técnica de suavização. Uma das técnicas de suavização mais simples é chamada de estimativa de Laplace (Correção Laplaciana).

Máquinas de Vetores de Suporte - *Support Vector Machines (SVMs)*

As máquinas de vetores de suporte vêm ganhando cada vez mais espaço dentro da comunidade de AM nos últimos anos. Seus resultados são muitas vezes superiores aos obtidos por outros algoritmos populares de aprendizado.

Máquinas de Vetores de Suporte - *Support Vector Machines (SVMs)*

As máquinas de vetores de suporte vêm ganhando cada vez mais espaço dentro da comunidade de AM nos últimos anos. Seus resultados são muitas vezes superiores aos obtidos por outros algoritmos populares de aprendizado.

> As máquinas de vetores de suporte (SVMs) são embasadas pela teoria de aprendizado estatístico (TAE) desenvolvido por Vapnik (1995) a partir dos estudos iniciados por Vapnik e Chervonenkis (1971).

SVMs Lineares

As SVMs surgiram por meio do emprego direto dos resultados obtidos pela Teoria de Aprendizado Estatístico. Primeiramente, vamos lidar com problemas linearmente separáveis.

SVMs Lineares

As SVMs surgiram por meio do emprego direto dos resultados obtidos pela Teoria de Aprendizado Estatístico. Primeiramente, vamos lidar com problemas linearmente separáveis.

As SVMs lineares com margens rígidas definem fronteiras lineares a partir de dados linearmente separáveis.

SVMs Lineares

As SVMs surgiram por meio do emprego direto dos resultados obtidos pela Teoria de Aprendizado Estatístico. Primeiramente, vamos lidar com problemas linearmente separáveis.

As SVMs lineares com margens rígidas definem fronteiras lineares a partir de dados linearmente separáveis.

Seja X um conjunto de treinamento com n objetos S e seus respectivos rótulos $y_i \in Y$, em que X constitui o espaço de entrada e $Y = \{-1, +1\}$ são as possíveis classes. Dizemos que X é linearmente separavél se é possível separar os objetos das classes + 1 e - 1 por um hiperplano.

SVMs Lineares

As SVMs surgiram por meio do emprego direto dos resultados obtidos pela Teoria de Aprendizado Estatístico. Primeiramente, vamos lidar com problemas linearmente separáveis.

As SVMs lineares com margens rígidas definem fronteiras lineares a partir de dados linearmente separáveis.

Seja X um conjunto de treinamento com n objetos S e seus respectivos rótulos $y_i \in Y$, em que X constitui o espaço de entrada e $Y = \{-1, +1\}$ são as possíveis classes. Dizemos que X é linearmente separavél se é possível separar os objetos das classes + 1 e - 1 por um hiperplano.

Classificadores que separam os dados por meio de um hiperplano são ditos lineares.

SVMs Lineares

Basicamente, o SVM é um algoritmo supervisionado que tenta criar uma linha (ou uma fronteira) que melhor separa os dados. Essa linha preta é o melhor Hiperplano que o algoritmo conseguiu criar, ou seja, o hiperplano que melhor separa as classes.

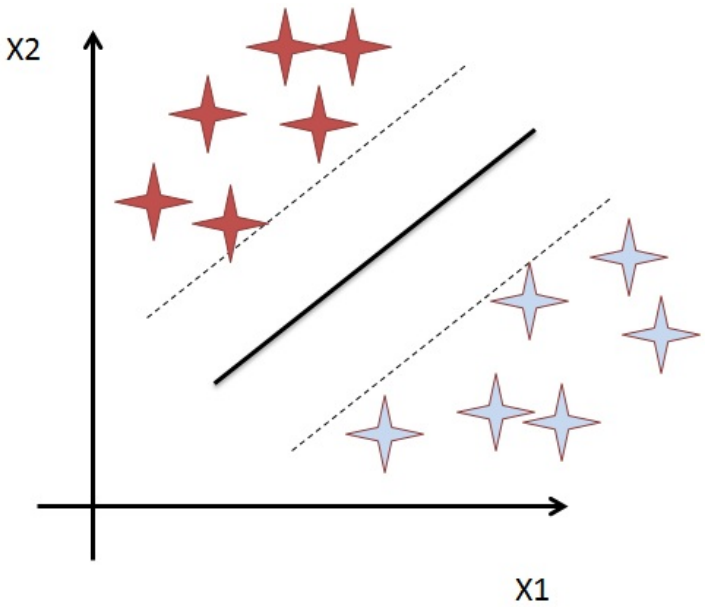


Figura : Hiperplano de Classificação.

SVMs Lineares

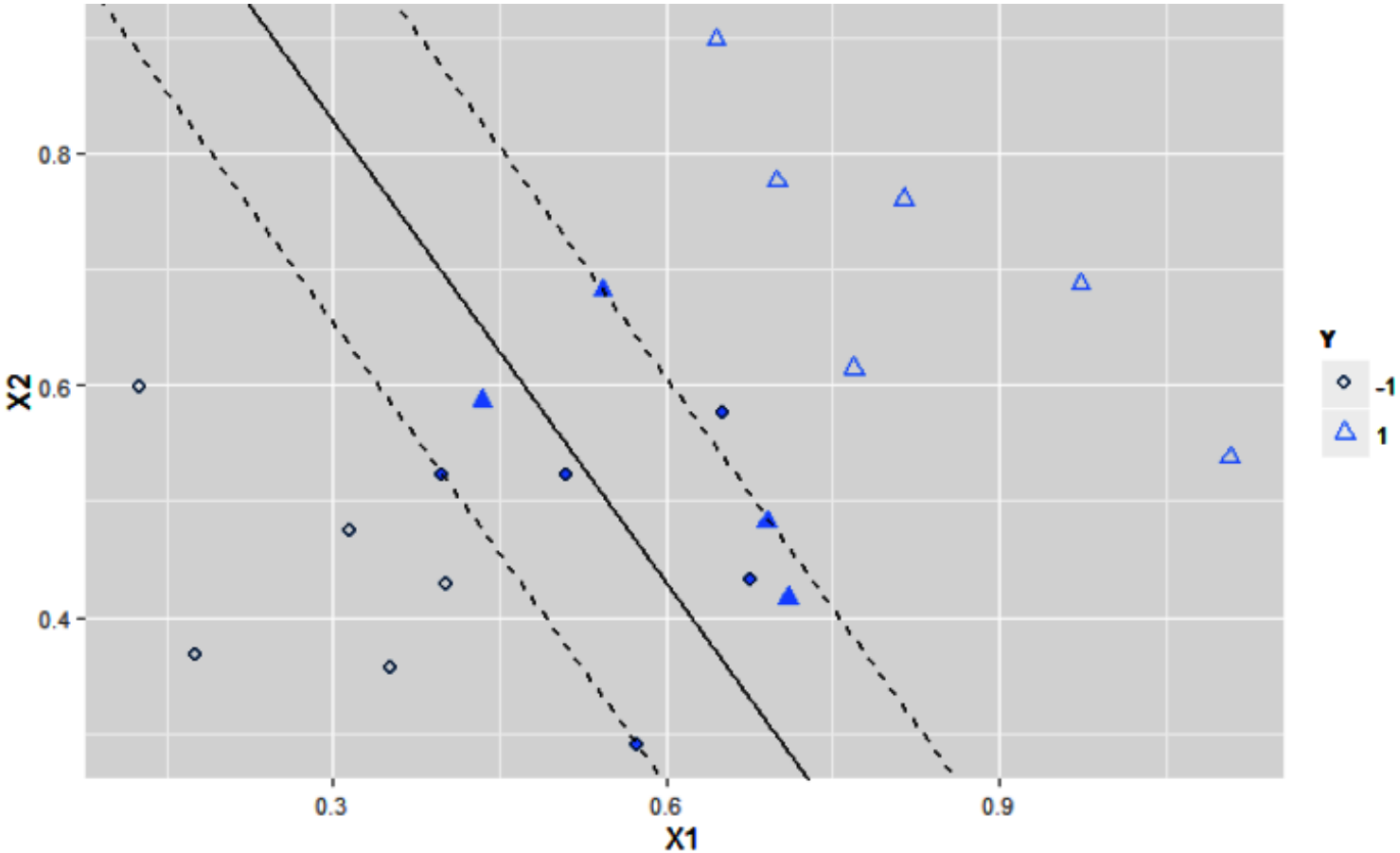


Figura : Exemplo de Hiperplano Separador Ótimo.

SVMs Não Lineares

Há inúmeros casos em que não é possível dividir de forma satisfatória os dados de treinamento por meio de um hiperplano.

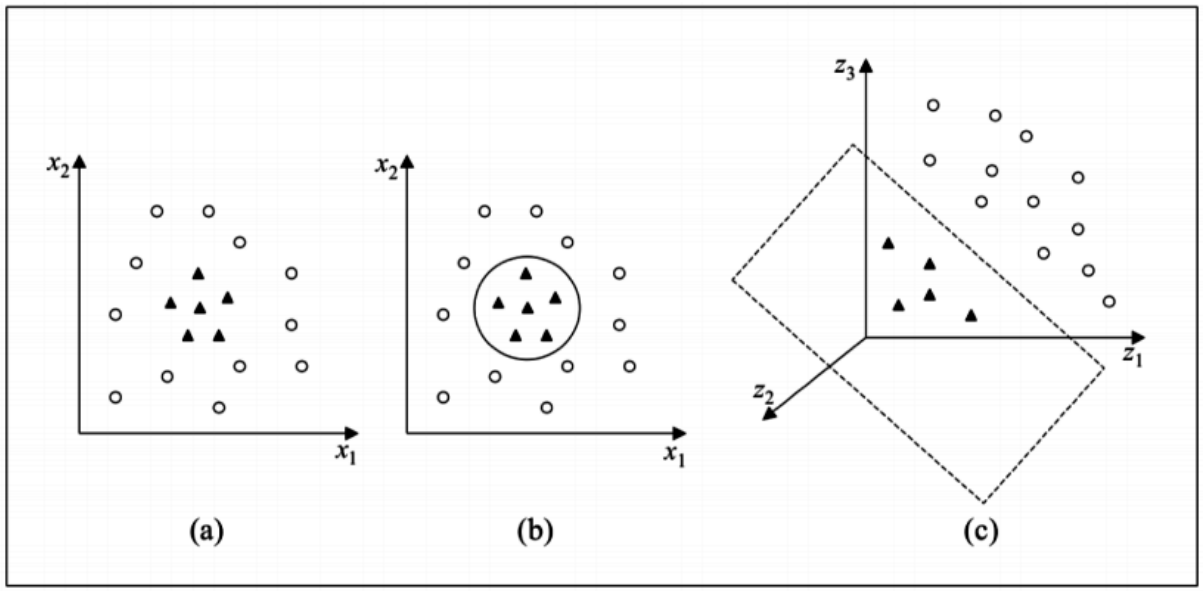


Figura : (a) Conjunto de dados não linear; (b) Fronteira não linear no espaço de entradas; (c) Fronteira linear no espaço de características - Lorena e Carvalho, 2007.

SVMs Não Lineares

Quando trabalhamos com SVM não lineares é comum utilizarmos funções denominadas kernels.

Um kernel K é uma função que recebe dois pontos x_i e x_j do espaço de entradas e computa o produto escalar desses dados no espaço de características (Herbrich, 2001).
Assim, tem-se:

$$K(x_i, x_j) = \langle h(x_i), h(x_j) \rangle . \tag{12}$$

A utilização dos kernels reside na simplicidade de seu cálculo e na sua capacidade de representar espaços abstratos.

SVMs Não Lineares

Alguns dos kernels mais utlizados na prática são os polinomiais, os de função de base radial, (*radial basis function* - RBF) e os sigmoidais.

Kernel Linear:

$$K(x_i, x_j) = \langle x_i, x_j \rangle \tag{13}$$

Kernel Polinomial:

$$K(x_i, x_j) = (\delta(x_i \cdot x_j) + \kappa)^d \tag{14}$$

SVMs Não Lineares

Kernel tangente hiperbólica (Sigmoidal):

$$K(x_i, x_j) = tanh(\delta(x_i \cdot x_j) + \kappa) \tag{15}$$

Kernel Radial Basis Function (RBF):

$$K(x_i, x_j) = exp(-\sigma \|x_i - x_j\|^2) \tag{16}$$

em que δ, κ, d e σ são os hiper-parâmetros que devem ser selecionados empiricamente.

SVMs Não Lineares

A obtenção de um classficador por meio do uso de SVM envolve a escolha de uma função kernel e dos parâmetros dessa função e da constante de regularização.

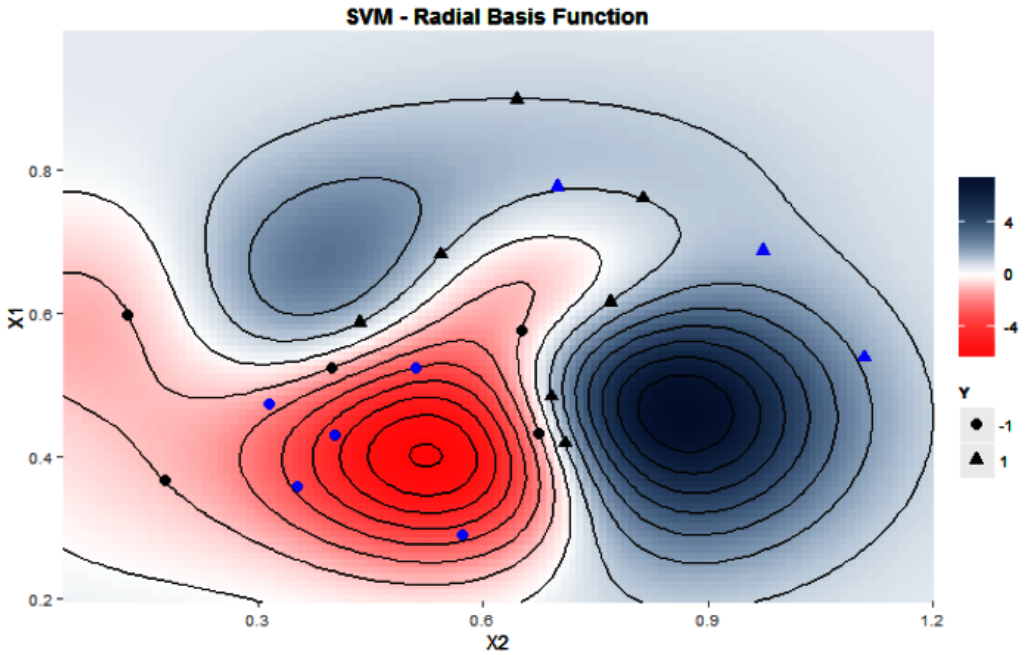


Figura : Exemplo Support Vector Machine - Kernel RBF.

SVM Vantagens e Desvantagens

Vantagens:

1) As SVMs são robustas diante de grande dimensão

SVM Vantagens e Desvantagens

Vantagens:

1) As SVMs são robustas diante de grande dimensão

2) A convexidade do problema de otimização implica na existência de um único mínimo global

SVM Vantagens e Desvantagens

Vantagens:

1) As SVMs são robustas diante de grande dimensão

2) A convexidade do problema de otimização implica na existência de um único mínimo global

3) O processo de classificação é rápido

SVM Vantagens e Desvantagens

Vantagens:

1) As SVMs são robustas diante de grande dimensão

2) A convexidade do problema de otimização implica na existência de um único mínimo global

3) O processo de classificação é rápido

4) Costumam apresentar uma alta precisão na predição de valores;

SVM Vantagens e Desvantagens

Vantagens:

1) As SVMs são robustas diante de grande dimensão

2) A convexidade do problema de otimização implica na existência de um único mínimo global

3) O processo de classificação é rápido

4) Costumam apresentar uma alta precisão na predição de valores;

Desvantagens:

1) O tempo de treinamento pode ser bem longo dependendo do número de atributos e da dimensionalidade dos dados.

SVM Vantagens e Desvantagens

Vantagens:

1) As SVMs são robustas diante de grande dimensão

2) A convexidade do problema de otimização implica na existência de um único mínimo global

3) O processo de classificação é rápido

4) Costumam apresentar uma alta precisão na predição de valores;

Desvantagens:

1) O tempo de treinamento pode ser bem longo dependendo do número de atributos e da dimensionalidade dos dados.

2) Os modelos fornecidos não são facilmente compreensíveis (interpretáveis).

Métricas para Classificação

A avaliação de um algoritmo de AM supervisionado é normalmente realizada por meio da análise do desempenho do preditor gerado por ele na rotulação de novos objetos, não apresentados previamente em seu treinamento (Monard e Baranauskas,2003).

Uma medida de desempenho que pode ser empregada na avaliação de um classificador $\hat{f}$ é a sua taxa de erro ou de classificações incorretos, em que $I(a) = 1$ se a é verdadeiro e 0 em caso contrário. Esse tipo de medida equivale ao uso da função de custo 0 - 1 relacionando os rótulos dos objetos às predições obtidas:

$$erro(\hat{f}) = \frac{1}{n} \sum_{i=1}^{n} I\left(y_i \neq \hat{f}(x_i)\right). \tag{17}$$

Métricas para Classificação

A taxa de erro varia entre 0 e 1, sendo que valores próximos de 0 são melhores. O complemento dessa taxa é chamado de acurácia do classificador conforme a equação (18):

$$ac(\hat{f}) = 1 - erro(\hat{f}) \tag{18}$$

Matriz de Confusão

A matriz de confusão também permite avaliar o desempenho de um classificador, no qual mostra o número de previsões corretas e incorretas em classe.

		Classificação				
		1	2	⋯	c	Total
Verdade	1	x_{11}	x_{12}	⋯	x_{1c}	x_{1+}
	2	x_{21}	x_{22}	⋯	x_{2c}	x_{2+}
	⋮	⋮	⋮	⋱	⋮	⋮
	c	x_{c1}	x_{c2}	⋯	x_{cc}	x_{c+}
	Total	x_{+1}	x_{+2}	⋯	x_{+c}	n

Figura : Matriz de Confusão.

Medidas de Desempenho

De acordo com Monard e Baranauskas (2003) temos as seguintes medidas de desempenho:

Taxa de erro na classe positiva - taxa de falsos negativos (TFN)

$$erro_{+}(\hat{f}) = \frac{FN}{VP + FN} \tag{19}$$

Taxa de erro na classe negativa - taxa de falsos positivos (TFP)

$$erro_{-}(\hat{f}) = \frac{FP}{FP + VN} \tag{20}$$

Taxa de erro total

$$erro(\hat{f}) = \frac{FP + FN}{n} \tag{21}$$

Medidas de Desempenho

Taxa de acerto - Acurácia

$$ac(\hat{f}) = \frac{VP + VN}{n} \tag{22}$$

Precisão

$$prec(\hat{f}) = \frac{VP}{VP + FP} \tag{23}$$

Sensibilidade - Taxa de verdadeiros positivos (TVP)

$$TVP(\hat{f}) = \frac{VP}{VP + FN} \tag{24}$$

Especificidade - taxa de acerto na classe negativa

$$esp(\hat{f}) = \frac{VN}{VN + FP} = 1 - TFP(\hat{f}) \tag{25}$$

Validação Cruzada

No método de validação cruzada k - *fold cross-validation*, o conjunto de exemplos é divido k subconjuntos de tamanho aproximadamente igual. Os objetos k - 1 partições são utilizados no treinamento de um preditor, o qual é então testado na partição testado na partição restante.

Esse processo é repetido k vezes, utilizando em cada ciclo uma partição diferente para teste.

O desempenho final do preditor é dado pela médida dos desempenhos observados sobre cada subconjunto de teste.

Validação Cruzada

O *k-fold cross validation* consiste em dividir a base em k pedaços. Para cada pedaço, estimamos o método sem a presença desta parte e verificamos o erro médio no pedaço não utilizado durante o treino. Abaixo, um exemplo de *6-fold cross validation*.

1	2	3	4	5	6
Treino	Validação	Treino	Treino	Treino	Treino

Tabela : Exemplo de validação cruzada -$k = 6$

Underfitting e Overfitting

Os algoritmos não podem generalizar demais, porém não podem ser específicos demais, isto é, precisamos de algoritmos equilibrados.

Underfitting: O modelo trabalha de modo insatisfatório nos dados de treinamento. Isso ocorre porque o modelo não consegue capturar o relacionamento entre os exemplos de entrada (geralmente denominado X) e os valores de destino (geralmente denominado Y) (alto viés).

Underfitting e Overfitting

Os algoritmos não podem generalizar demais, porém não podem ser específicos demais, isto é, precisamos de algoritmos equilibrados.

Underfitting: O modelo trabalha de modo insatisfatório nos dados de treinamento. Isso ocorre porque o modelo não consegue capturar o relacionamento entre os exemplos de entrada (geralmente denominado X) e os valores de destino (geralmente denominado Y) (alto viés).

Overfitting: O ocorre quando os dados de treinamento quando você percebe que ele trabalha de modo satisfatório nos dados de treinamento, mas não nos dados de avaliação. Isso acontece porque o modelo está memorizando os dados reconhecidos e não consegue fazer a generalização nos exemplos não vistos.

Underfitting e Overfitting

A precisão nos dados de treinamento e de teste pode ser insatisfatória porque o algoritmo de aprendizagem não tem dados suficientes para serem aprendidos. Melhore o desempenho fazendo o seguinte:

Aumentar a quantidade de exemplos de dados de treinamento.

Aumentar o número de passagens nos dados de treinamento existentes.

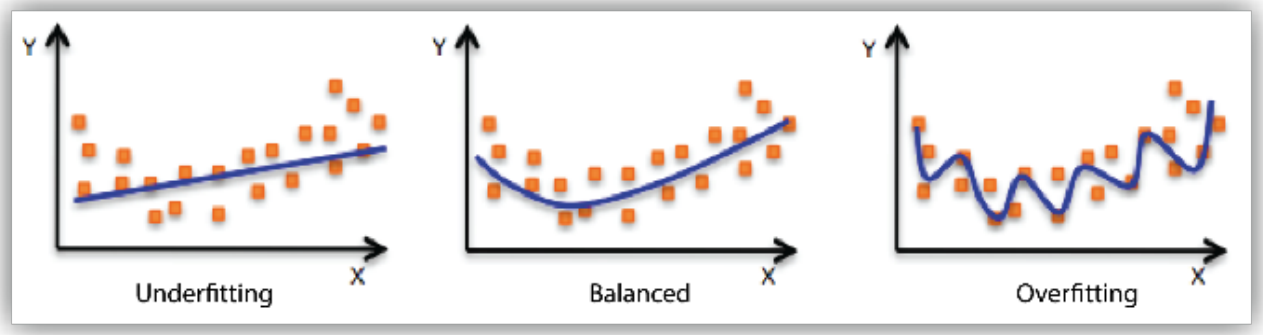


Figura : Exemplos de ajustes de modelos.

Referências

1) Bhatia, A.; Chiu, Y-Wei(David). Machine Learning with R Cookbook. Packt Publishing Limited, Second Edition, - Birmingham, 2017.

2) Demsár, J. Statistical comparisons of classifiers over multiple datasets. Journal of Machine Learning Research, 7:1-30, 2006.

3) Drucker, H., Wu, D.; Vapnik, V. (1999). Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 10, 1048-1054.

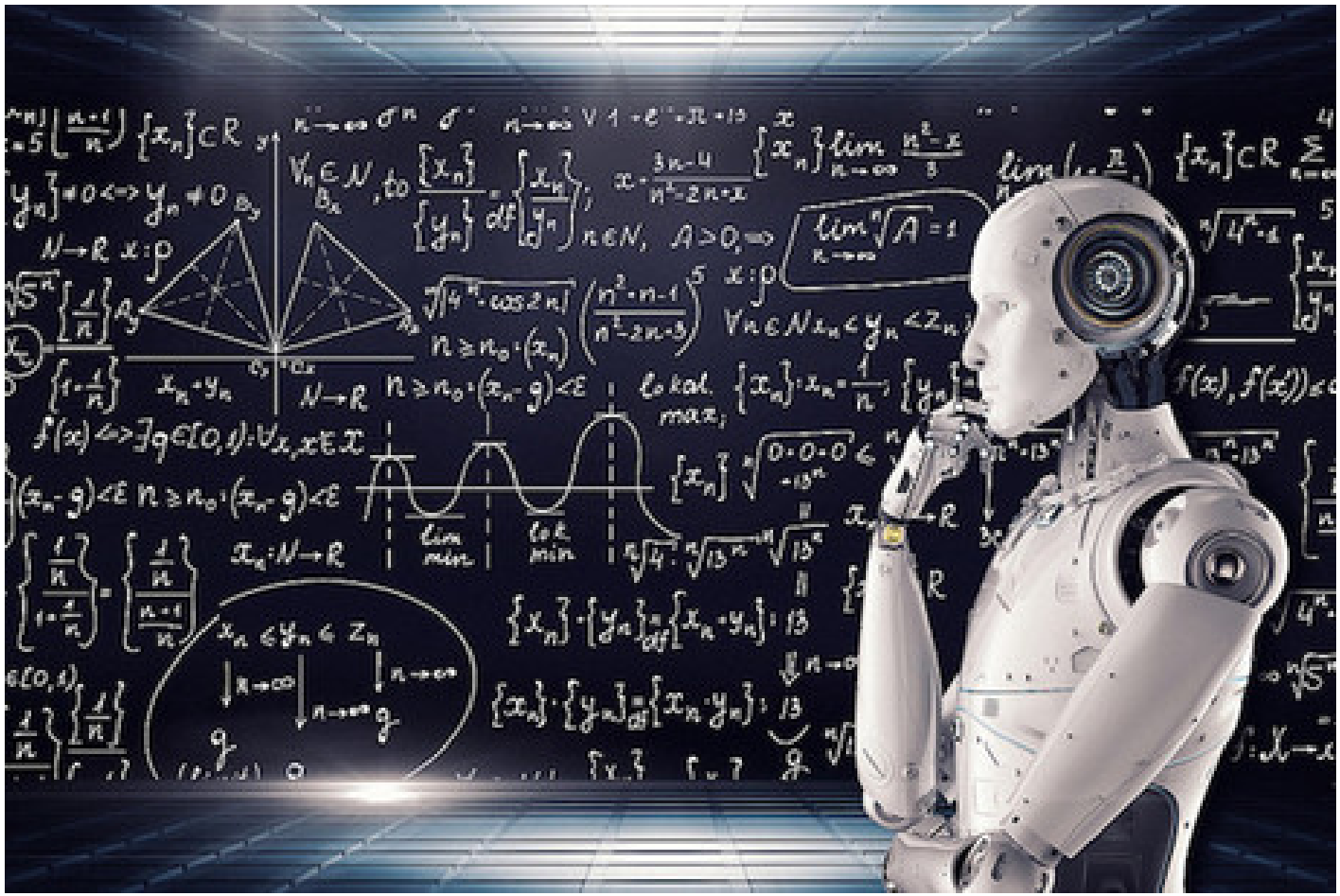
4) FACELI, Katti. et al. Inteligência Artificial: uma abordagem de aprendizagem de máquina. Rio de Janeiro: LTC, 2011.

Referências

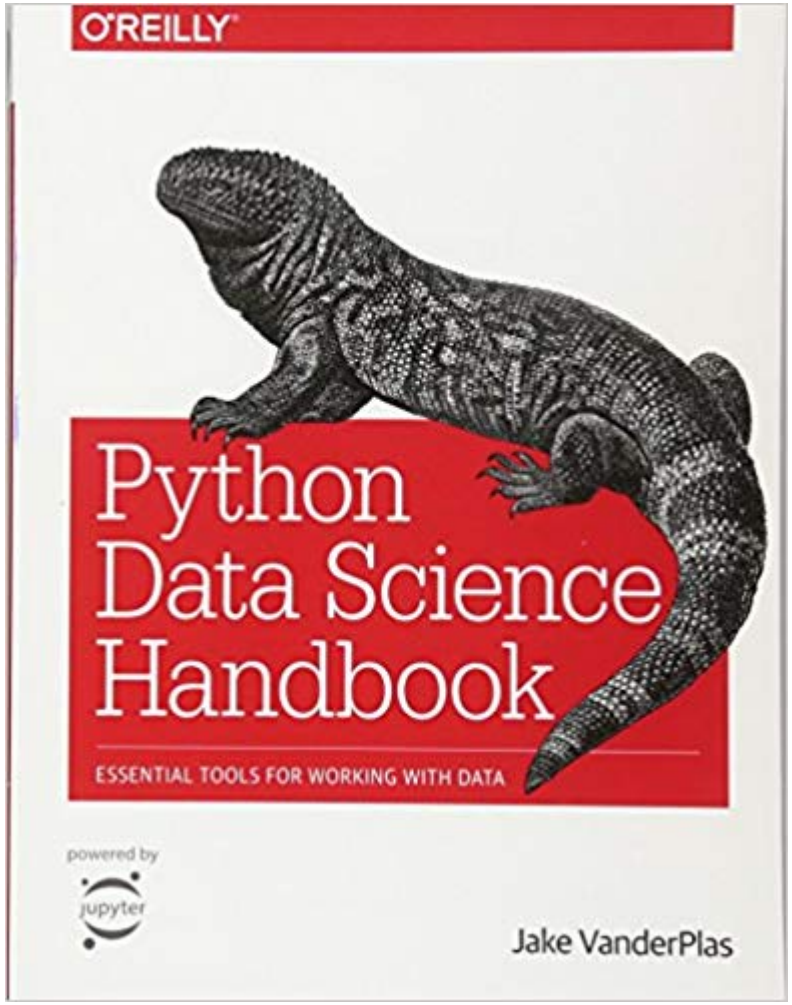
5) Frank E, Hall M, Trigg L, Holmes G, Witten IH: Data mining in bioinformatics using Weka. Bioinformatics 2004, 20:2479-81.

6) Fukunaga, K.; Narendra, P. M. A branch and bound algorithm for computing k-nearest neighbors. IEEE Transactions on Computers, v. 100, n. 7, p.750-753, 1975.

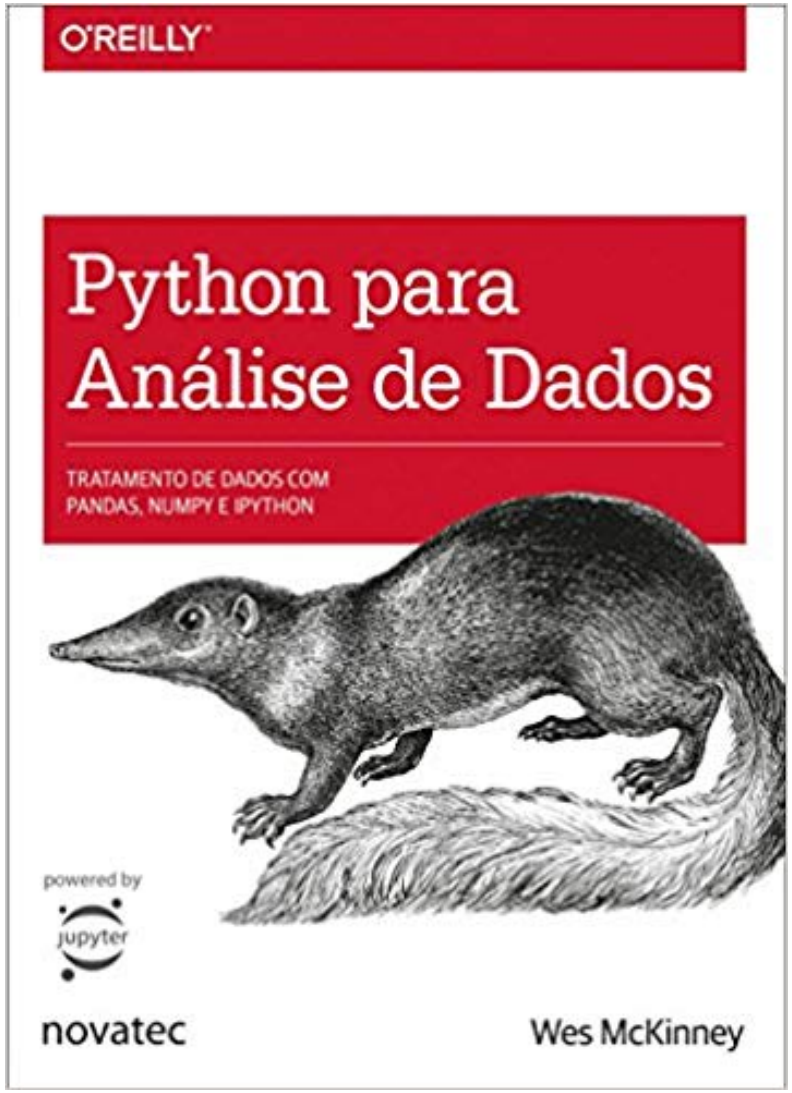
OBRIGADO PELA ATENÇÃO



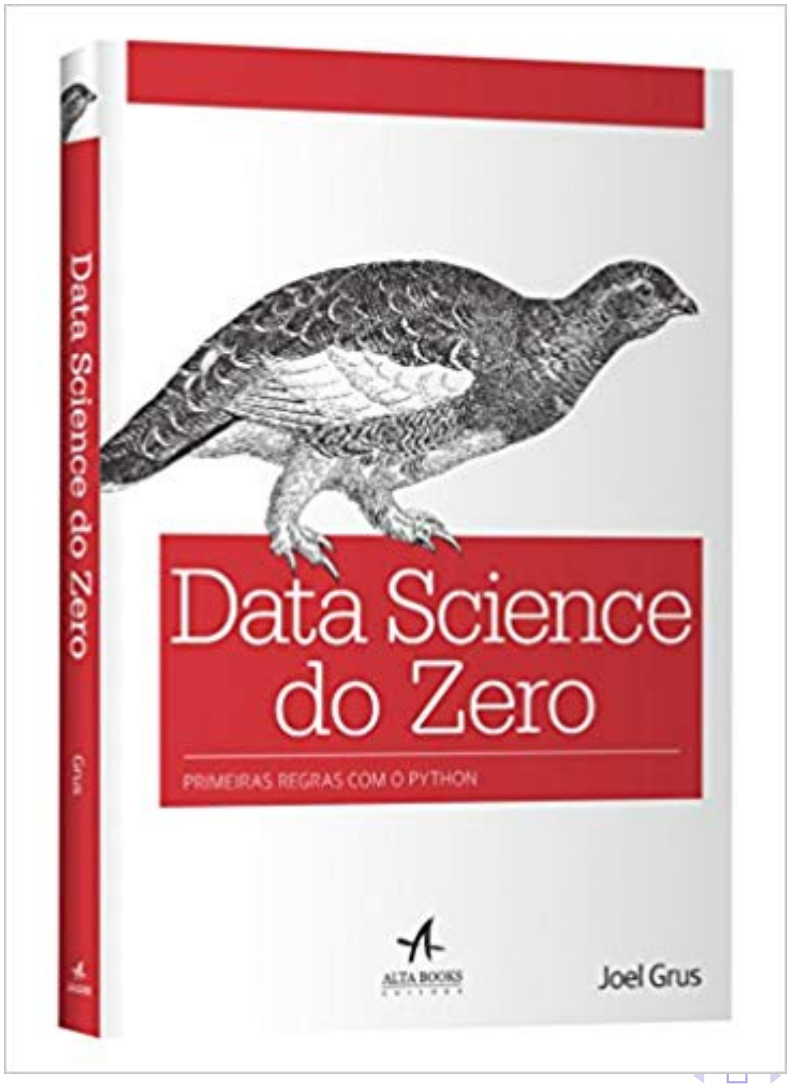
Alguns Livros interessantes...



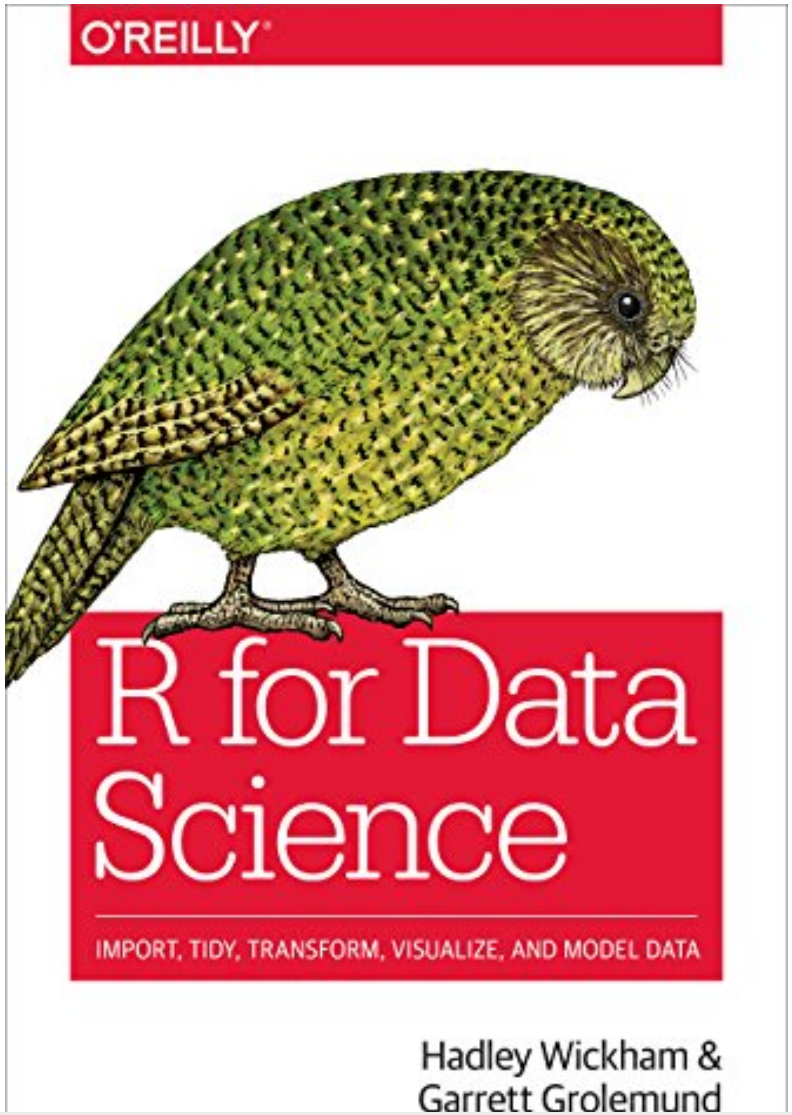
Alguns Livros interessantes...



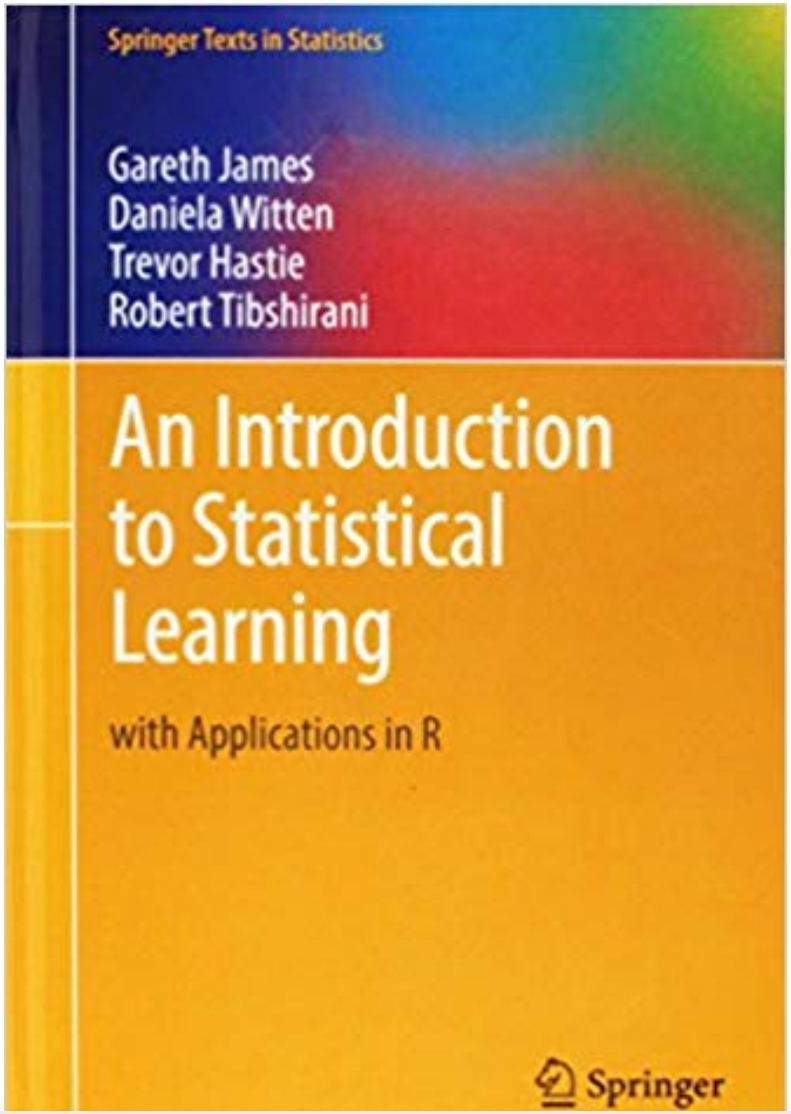
Alguns Livros interessantes...



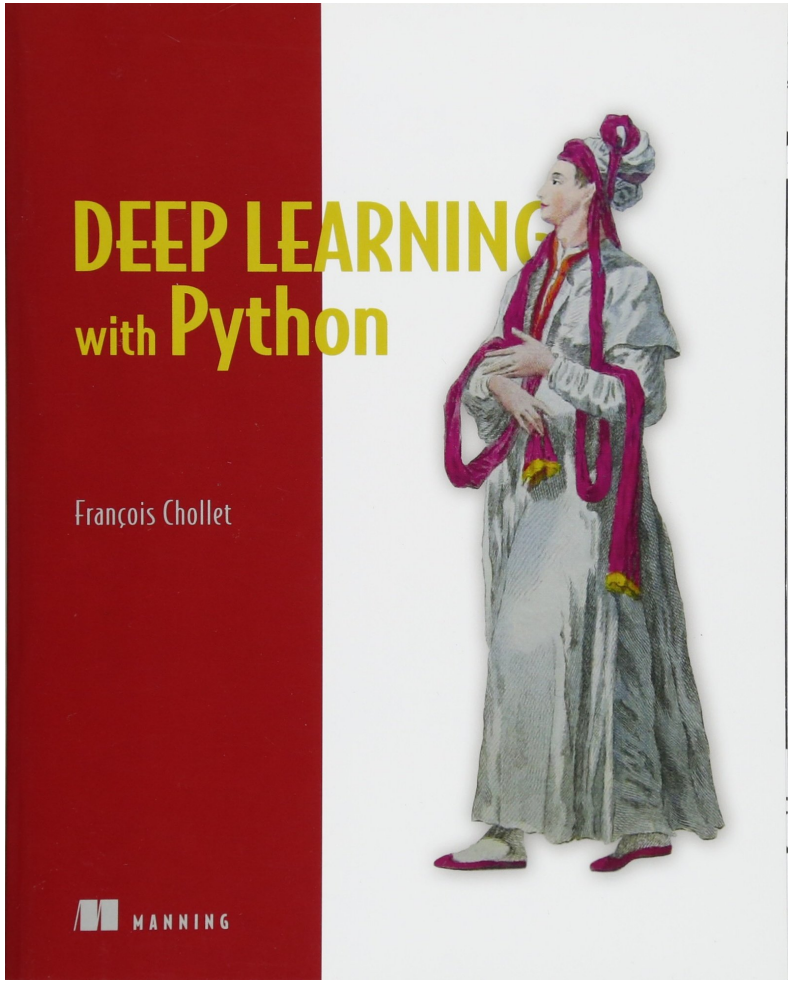
Alguns Livros interessantes...



Alguns Livros interessantes...



Alguns Livros interessantes...



Alguns Livros interessantes...