



# Curso de Extensão



## Ciência dos Dados em Administração

*Por Gustavo Alexandre*



# Gustavo Alexandre

- DBA na UFF ([STI](#))
- E-mail: [gassantos@id.uff.br](mailto:gassantos@id.uff.br)
- LinkedIn: <https://linkedin.com/in/gassantos>
- GitHub: <https://github.com/gassantos>
- Curso: <https://github.com/curso-extensao-uff>

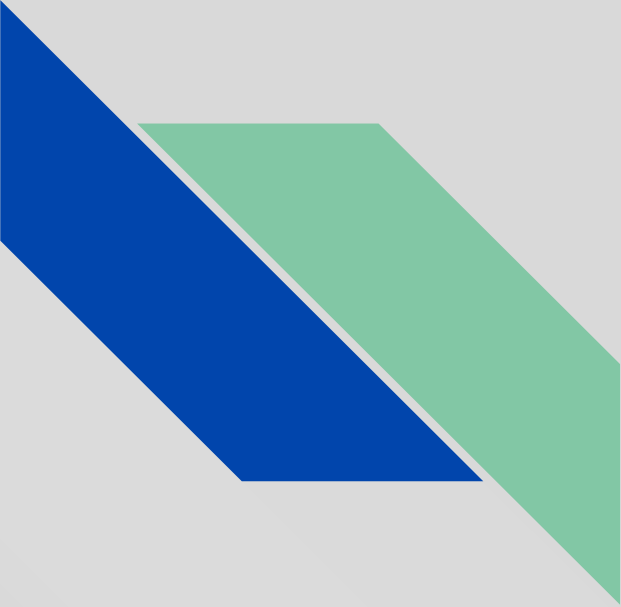
# Aprendizado de Máquina





# Agenda

- 1. Mercado**
- 2. Aprendizado de Máquina (AM)**
- 3. Cenários de Aprendizado de Máquina**
- 4. Processo de Aprendizado de Máquina**
- 5. Aprendizado Supervisionado**
- 6. Técnicas**
- 7. Avaliação**



# Mercado

**“Há três tipos de mentiras: as mentiras, as mentiras descabeladas, e as estatística”**

*Benjamin Disraeli*

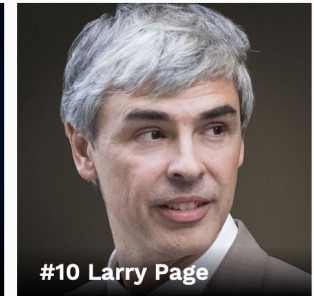
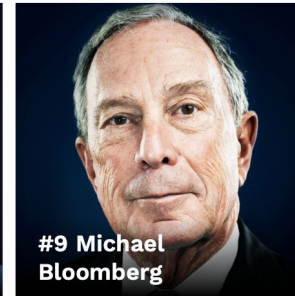
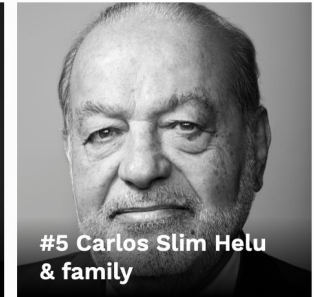
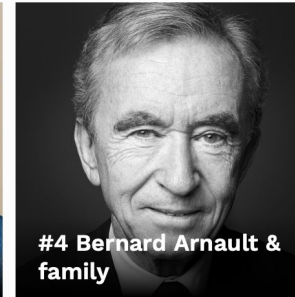


# As Maiores Empresas do Mundo em 2019

Segundo a Forbes, ***Das trinta maiores empresas do mundo, nove são empresas de tecnologia:*** [Apple](#), [AT&T](#), [Samsung](#), [Microsoft](#), [Alphabet](#), [Verizon](#), [China Mobile](#) e [Amazon](#).

Fonte: [Forbes Global 2000 \(2019\)](#)

# Os homens mais ricos do Mundo em 2019



**Fonte:** Forbes Global 2000 (2019)

# Os homens mais ricos do Mundo em 2015

Top 100<sup>[1]</sup> [\[ edit \]](#)

No. ↕	Name ↕	Citizenship ↕	Age ↕	Net Worth USD billion ↕	Source(s) of wealth ↕
1	Bill Gates	United States	59	79.20	Microsoft
2	Carlos Slim Helu	Mexico	75	77.10	Telecom
3	Warren Buffett	United States	84	72.70	Berkshire Hathaway
4	Amancio Ortega	Spain	78	64.50	Zara
5	Larry Ellison	United States	70	54.30	Oracle
6	Charles Koch	United States	79	42.90	Diversified
6	David Koch	United States	74	42.90	Diversified
8	Christy Walton	United States	60	41.70	Wal-Mart
9	Jim Walton	United States	67	40.60	Wal-Mart
10	Liliane Bettencourt	France	92	40.10	L'Oréal
11	<a href="#">Alice Walton</a>	United States	65	39.40	<a href="#">Wal-Mart</a>
12	<a href="#">S. Robson Walton</a>	United States	71	39.10	Wal-Mart
13	<a href="#">Bernard Arnault</a>	France	66	37.20	<a href="#">LVMH</a>
14	<a href="#">Michael Bloomberg</a>	United States	73	35.50	<a href="#">Bloomberg LP</a>
15	<a href="#">Jeff Bezos</a>	United States	51	34.80	<a href="#">Amazon.com</a>
16	<a href="#">Mark Zuckerberg</a>	United States	30	33.40	<a href="#">Facebook</a>
17	<a href="#">Li Ka-shing</a>	Hong Kong	86	33.30	Diversified
18	<a href="#">Sheldon Adelson</a>	United States	81	31.40	Casinos
19	<a href="#">Larry Page</a>	United States	41	29.70	<a href="#">Google</a>
20	<a href="#">Sergey Brin</a>	United States	41	29.20	Google

Fonte: Forbes Global 2000 (2015)



**Apple**





# Microsoft



Microsoft



# America Movil



# Samsung



# Facebook





Google



Biggest Tech Company Ever

Alphabet Inc.



is for Google



# O Case Amazon





# Transformação Digital

É o uso da tecnologia para resolver problemas tradicionais baseado-se em soluções digitais a fim de promover eficiência e automação aos procedimentos e atividades dos processos de negócio [[Christian Matt \*et al.\*, 2014](#)]





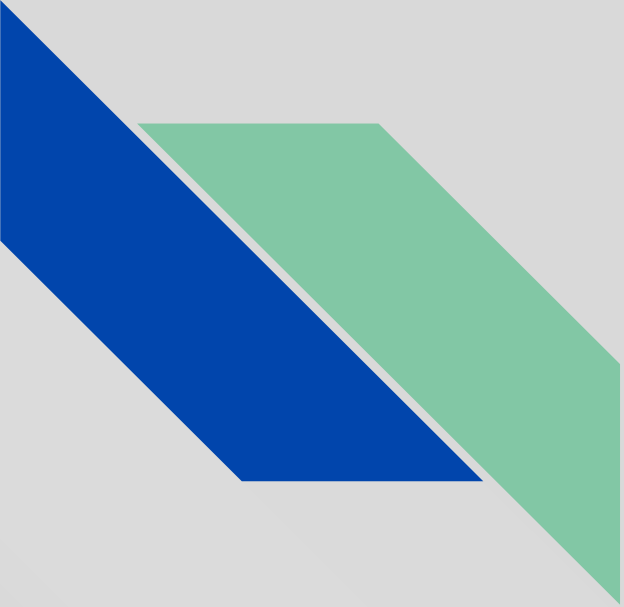
# Transformação de Negócio

É o processo de “reestruturação fundamental” dos sistemas, dos processos, das pessoas e da tecnologia em toda uma empresa ou unidade de negócio, para alcançar melhorias mensuráveis em eficiência, eficácia e satisfação das partes interessadas [[Cruise, 2017](#)]



## Leitura Complementar

- ❑ Estratégia Digital do Governo Federal 2017
- ❑ Information Economy Report 2015
- ❑ Digital Transformation
- ❑ Digital Transformation Strategies
- ❑ Industry 4.0
- ❑ Business Data Mining - A Machine Learning Perspective
- ❑ Business Intelligence and Analytics: From Big Data to Big Impact



# Aprendizado de Máquina

**"Os erros causados por dados inadequados são muito menores do que aqueles devido à sua falta"**

*Charles Babbage*

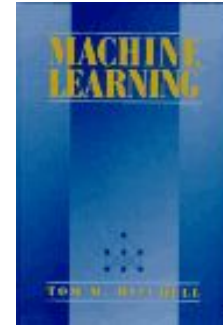
# Conceitos

Segundo Arthur Samuel (1959): “**área de estudo que concede aos computadores a habilidade de aprender sem serem programados explicitamente.**” [[Samuel, 1959](#)]



# Conceitos

Tom Mitchell (1998): “Um programa de computador aprende com a experiência  $E$  em relação a tarefa  $T$  e alguma medida de desempenho  $P$ , se seu desempenho em  $T$ , medido por  $P$ , melhora com a experiência  $E$ .”  
[[Mitchell, 1998](#)]





# Conceitos

O **Aprendizado de Máquina (AM)** explora o estudo e a construção de algoritmos que podem aprender sobre **dados** e fazer previsões



# Tipos de Aprendizado de Máquina

- Supervisionado
- Não-Supervisionado
- Semi-Supervisionado
- Profundo
- Por Reforço



# Aprendizado Supervisionado

- **(*supervised learning*)** visa construir um modelo estatístico a partir de um conjunto de dados que contém as entradas e as saídas desejadas (rotuladas):
  - Classificação
  - Regressão





# Aprendizado Não-Supervisionado

- ***(unsupervised learning)*** visa construir um modelo estatístico a partir de um conjunto de dados que contém apenas as informações de entradas e nenhuma identificação de saída (não rotulada):
  - Agrupamento
  - Sistemas de Recomendação
  - Filtragem



# Aprendizado Semi-Supervisionado

- (***semi-supervised learning***) visa construir um modelo estatístico a partir de um conjunto de dados de treinamento incompletos, em que uma parte da amostra possui rótulos e a outra não (rotulados e não-rotulados)



# Aprendizado Profundo

- (***deep learning***) compreende o uso das redes neurais artificiais em grandes volumes de dados (*big data*), ampliando continuamente sua capacidade de aprendizado, à medida que mais dados são processados
  - Visão Computacional
  - Reconhecimento de voz



# Aprendizado por Reforço

- (***reinforcement learning***) compreende a técnica de aprendizado interativo sobre a forma como agentes inteligentes (***multi-agent systems***) aprendem a agir em determinados ambientes, de modo a maximizar a noção de recompensa perante a execução das tarefas



## Leitura Complementar

- ❑ Introdução ao Aprendizado de Máquina - Genio  
|LTC
- ❑ Livros:
  - ❑ *Deep Learning*
  - ❑ Python Data Science Handbook
- ❑ Cursos:
  - ❑ EdX - Principles of Machine Learning
  - ❑ Google - Machine Learning Crash Course



# Cenários de Aprendizado de Máquina

**"Os fatos não deixam de existir apenas porque são ignorados"**

*Aldous Huxley*

# Exemplos

Detecção de spam

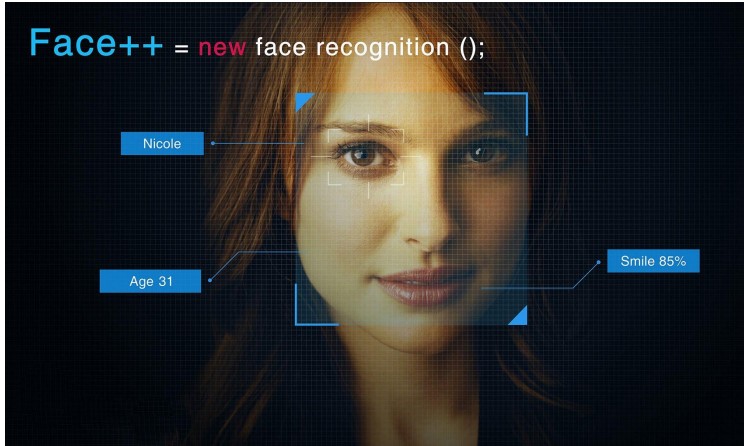


Reconhecimento de Voz

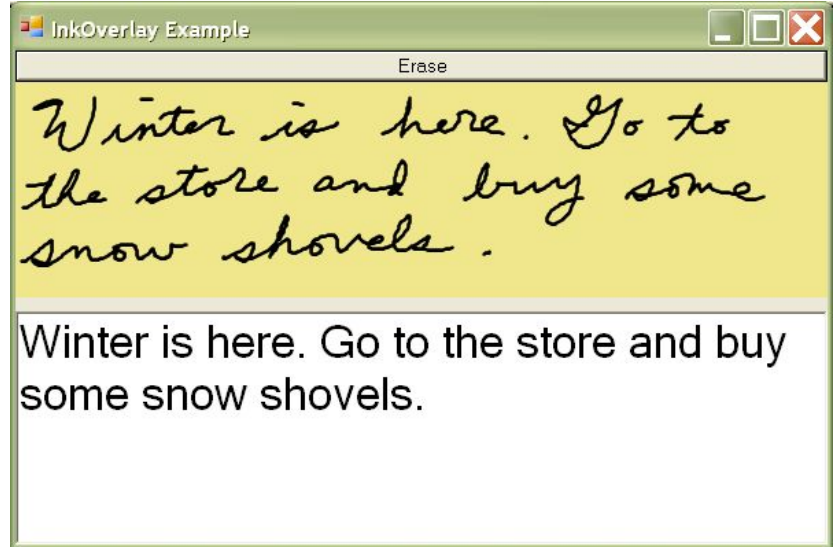


# Exemplos

## Reconhecimento de imagens



## Reconhecimento de caracteres



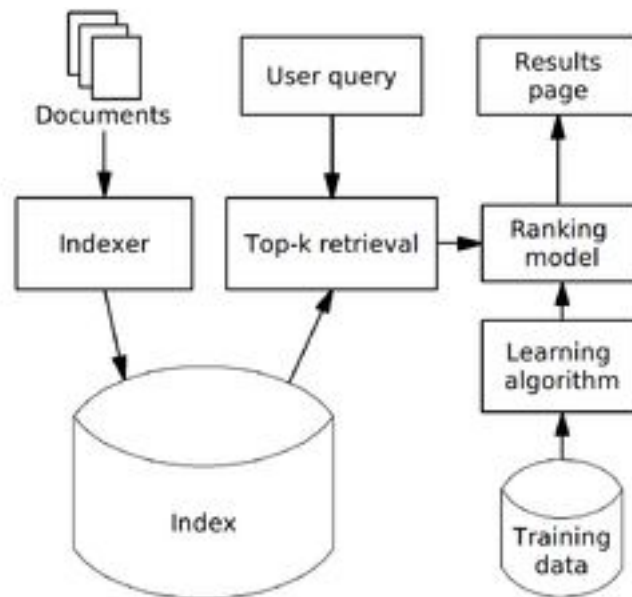


# Exemplos

## Tradução Automática



## Learning to rank





## Identifique o problema de AM

- ❖ Dado um exame, como identificar se um tumor é **benigno ou maligno com base no seu tamanho e idade do paciente?**



## Identifique o problema de AM

- ❖ Dado um conjunto de dados sobre o tamanho de casas no mercado imobiliário, como vamos **prever o preço de casas**, já que algumas instâncias foram atribuídas como padrão A, B e C?



## Identifique o problema de AM

- ❖ Dada uma imagem de homem ou mulher, como podemos **prever a sua idade com base em dados da imagem?**



## Identifique o problema de AM

- ❖ Dada uma coleção de milhares de pesquisas em uma universidade, como podemos encontrar uma maneira automática de **agrupar estas pesquisas que são de alguma forma semelhantes** por algumas variáveis, tais como a frequência das palavras, frases e contagem de páginas?



# Leitura Complementar

- ❏ Google AI - Education
- ❏ DSA - Casos de Uso de AM



# Processo de Aprendizado de Máquina

**"Há três tipos de mentiras: as mentiras, as mentiras descabeladas, e as estatísticas"**  
*Benjamin Disraeli*



# Aprendizado Supervisionado

- A máquina recebe as saídas identificadas
- Dois tipos (tarefas):
  - **Classificação**: prediz valor discreto
  - **Regressão**: prediz valor contínuo

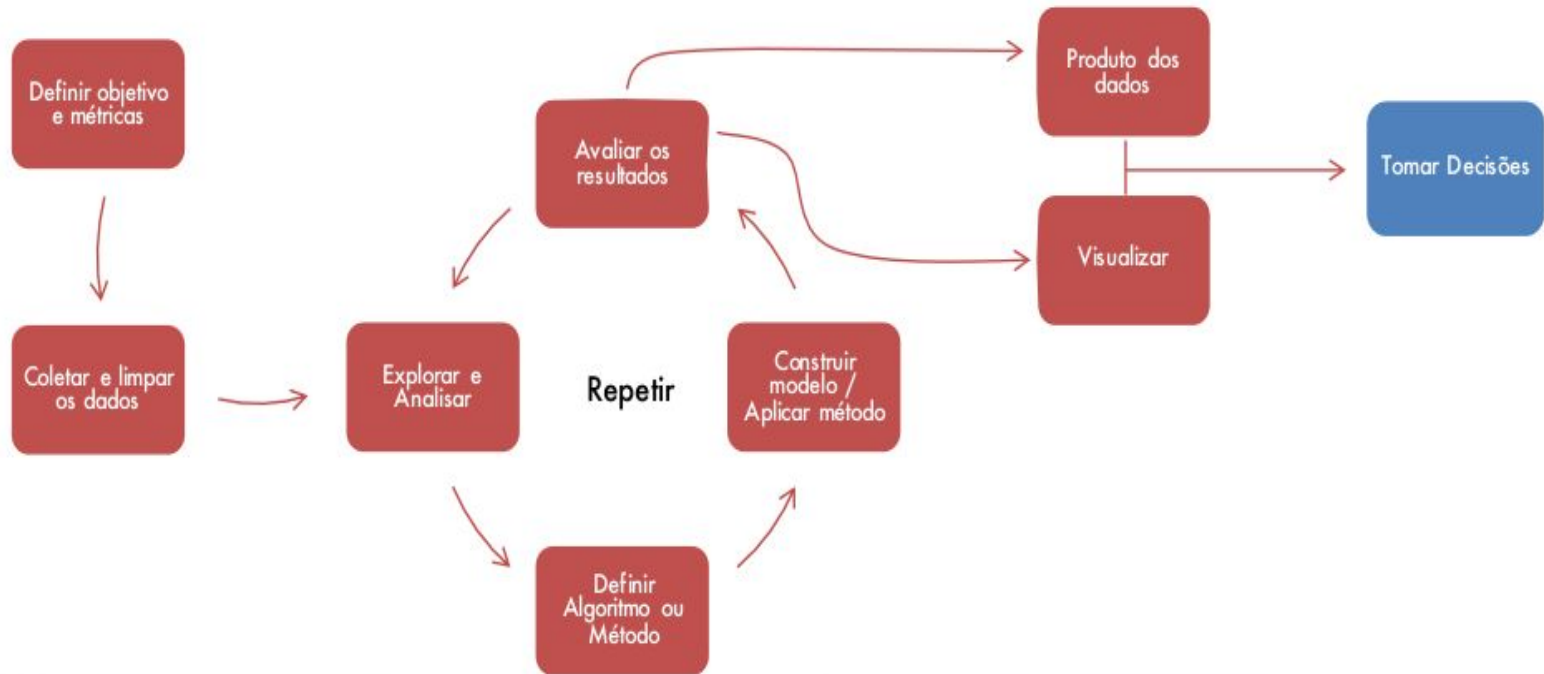




# Terminologia

- Conjunto de Treinamento
- Conjunto de Teste
- Conjunto de Validação (Produção, *Deploy*)
- **Alvo** (*Target*,  $\mathbf{y}$ ) - Vetor de Classes
- **Atributos** (*Features*,  $\mathbf{X}$ ) - Matriz de Variáveis
- **Modelo** (Técnica)
- **Algoritmo** de Aprendizado

# Fluxo dos Dados

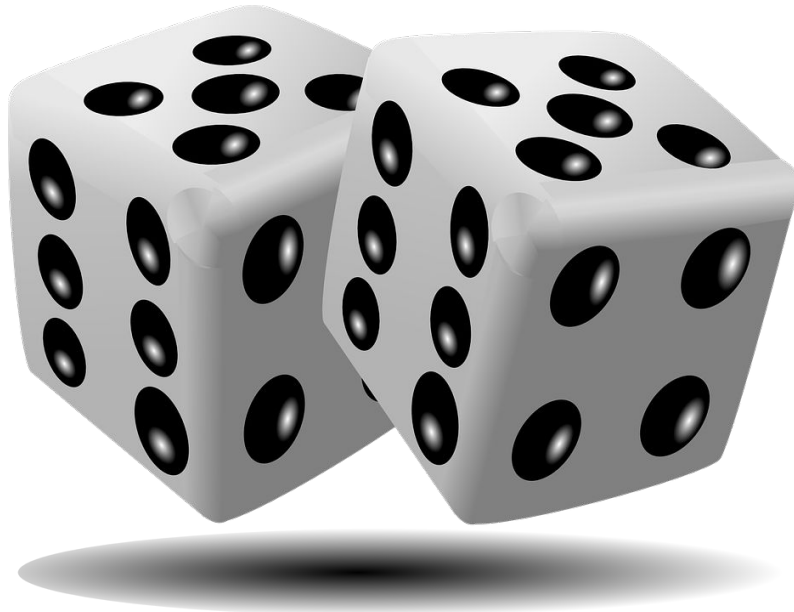




# Processo de Aprendizado de Máquina

- 1) Carregar os **dados**
- 2) Explorar os **dados**
- 3) Particionar os **dados**
  - a) **Treinamento**
  - b) **Teste**
- 4) Modelo => Ajustar os atributos (**model.fit**)
- 5) Modelo => Prever alvo (**model.predict**)

# Dados





## Carregando os Dados em *R*

- dados <- data(mydata)
- dados <- read.table("mydata.txt")
- dados <- read.csv("mydata.csv")
- dados <- read.xls("mydata.xls")

Fonte: [Data Import | R Import](#)



# Explorando os Dados em *R*

- **head**(dados)
- **nrow**(dados)
- **summary**(dados)
- **unique**(dados)
- **missing**(dados)



## Particionando os Dados em *R*

### a) Treinamento

- **treino** <- particion(dados, percentual)

### b) Teste

- **teste** <- dados[-treino]



## Ajustando os Dados em *R*

- “**Fittar**” os dados (*fitting*) é o procedimento de ajuste de dados ao modelo, analisando a precisão do ajuste. Podem ser usadas técnicas de equações matemáticas e métodos não paramétricos, para modelar os dados obtidos:
  - modelo <- **model.fit** (alvo, treino)





## Prevendo com os Dados em *R*

- “**Predição**” com dados (*predicting*) é o procedimento de analisar dados para fazer previsões. Geralmente, usa-se análises estatísticas e técnicas de AM para criar um modelo capaz de prever eventos futuros:
  - resultado <- **model.predict** (modelo, teste)



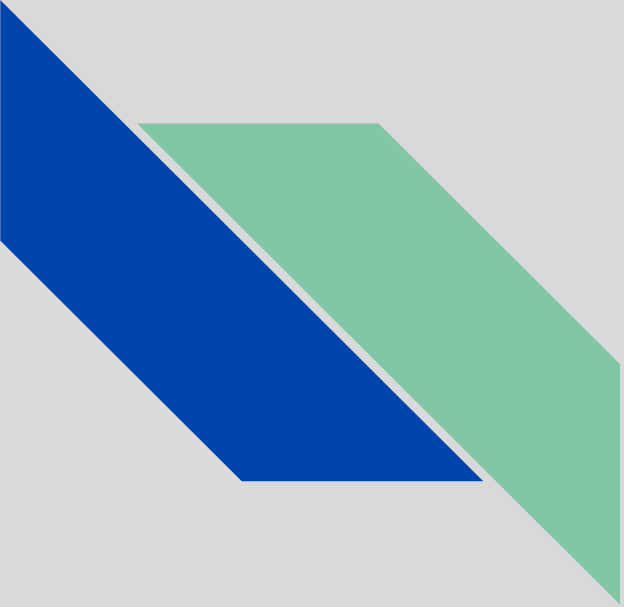
## Procedimento em AM Supervisionado

1. Identificar uma necessidade de negócio que possa ser resolvida com os dados disponíveis
2. Traduzir essa necessidade em um problema de AM Supervisionado (**classificação** ou **regressão**)
3. Adequação de rótulos aos dados históricos
4. Adequação de valores aos dados históricos



## Leitura Complementar

- ❑ Kaggle - *Introduction to Machine Learning in R*
- ❑ Livro - An Introduction to Machine Learning with R
- ❑ Curso - *Introduction to TensorFlow*
- ❑ Plataforma - *Google Colaboratory*
- ❑ Plataforma - Jupyter Hub



# Aprendizado Supervisionado (Classificação)

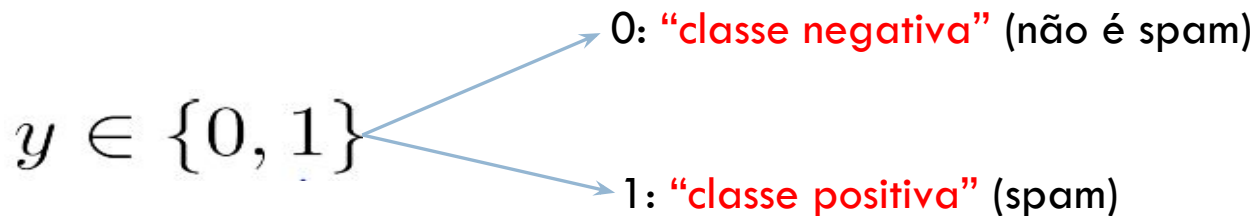
**“No futuro, o pensamento estatístico será tão necessário para a cidadania eficiente como saber ler e escrever” *H.G.Wells***



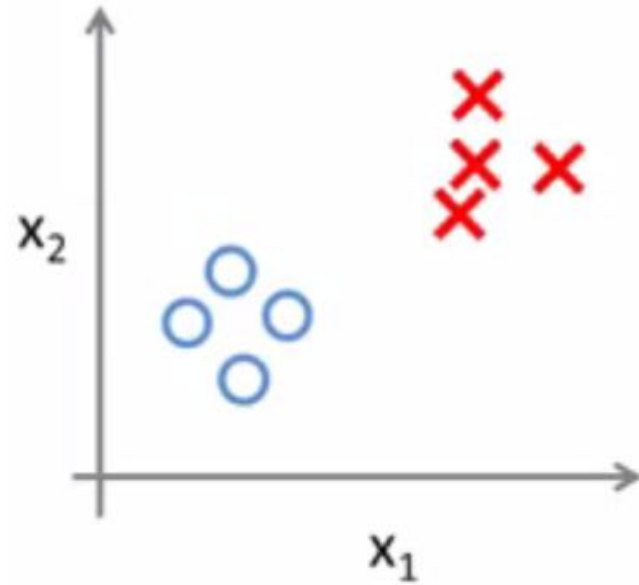
# Classificação Binária

## ● Exemplos

- **Email:** spam/ham (not spam)?
- **Transações financeiras:** fraudulenta/legítima?
- **Tumor:** maligno/benigno?



# Abordagem Binária

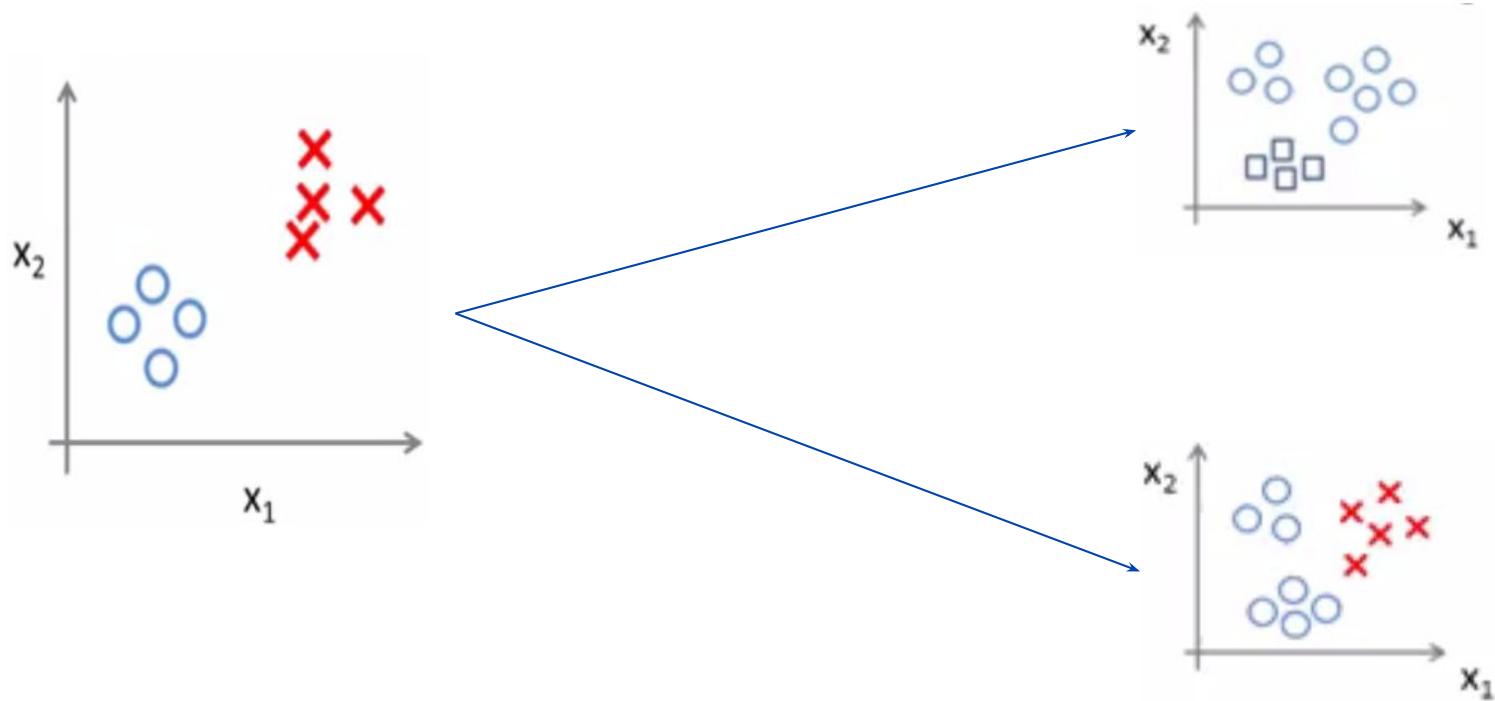




## Procedimento Binário

- Em geral para um problema de classificação com duas classes, os passos são:
  - Treinar um classificador para as classes com os dados de treinamento
  - Para fazer a classificação de um novo exemplo  $x$ , seleccionar os dados de teste

# Procedimento Binário

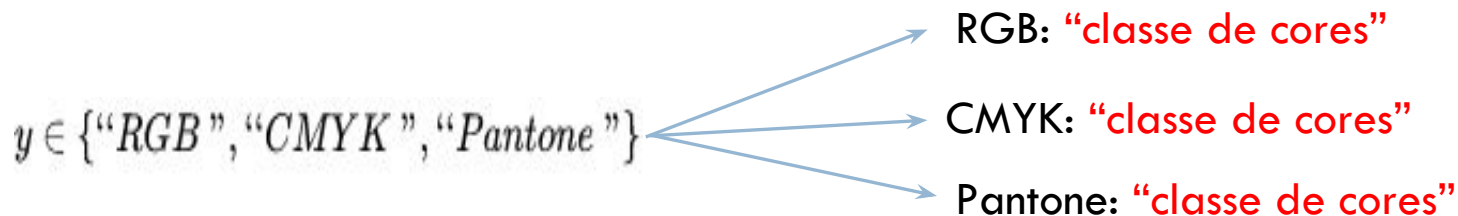




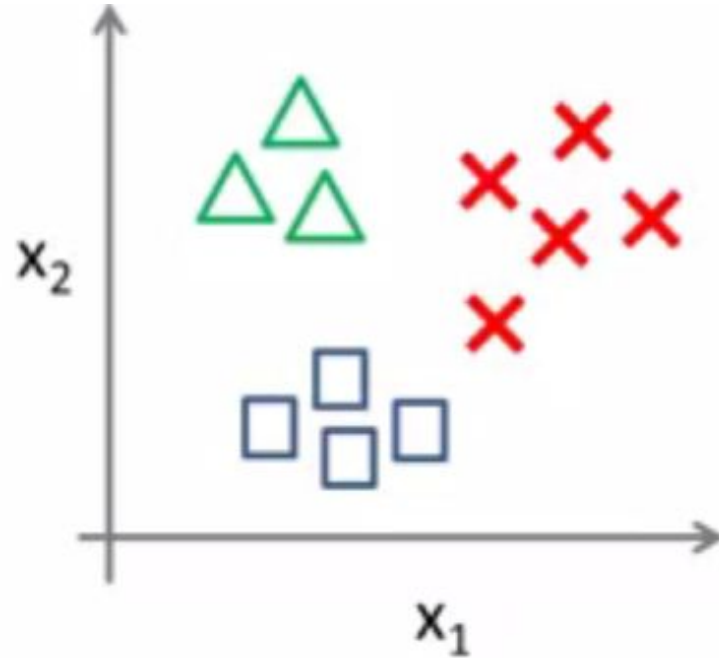
# Classificação Multiclasse

## ● Exemplos

- **Perfil político:** Liberal, Conservador ou Socialista?
- **Etnia/Raça:** Negro, Branco, Amarelo ou Indígena?
- **Clima:** ensolarado, nublado ou chuvoso?



# Abordagem Multiclasse

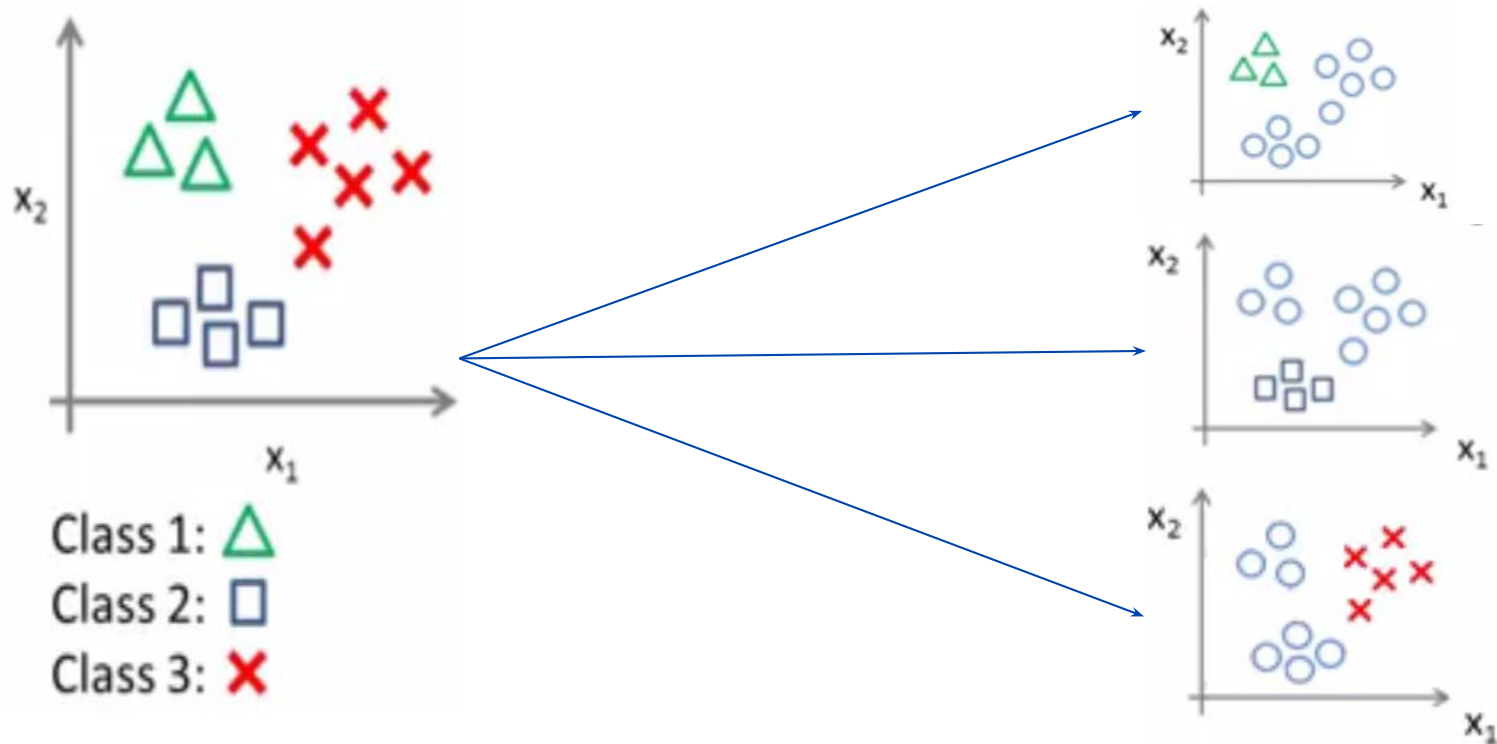




# Procedimento Multiclasse

- Em geral, para um problema de classificação com  $n$  classes, os passos são:
  - Treinar um classificador para cada uma das  $n$  classes
  - Para fazer a classificação de um novo exemplo  $x$ , seleccionar a classe que maximiza a hipótese correspondente

# Procedimento Multiclasse





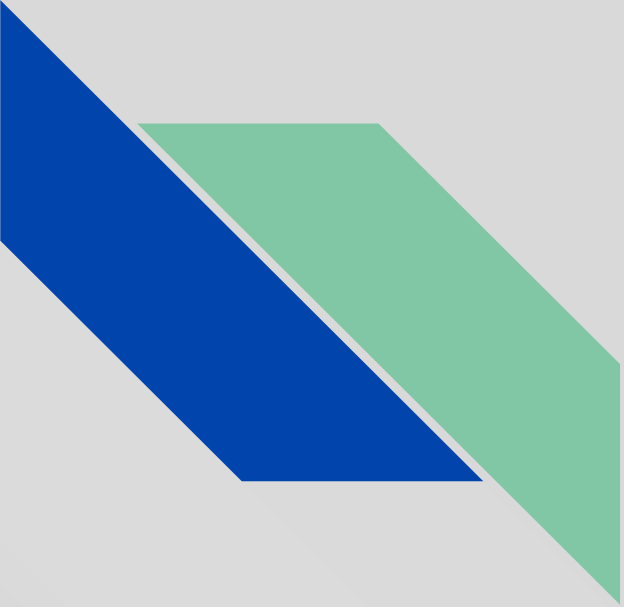
## Estudo de Caso p/ Multiclasse

- **Organização de um portal de notícias:** esportes, humor, política
- **Diagnose médica:** alergia, resfriado, dengue
- **Religião:** católica, protestante, espírita ou pentecostal



# Leitura Complementar

- ❏ Coursera - *Machine Learning: Classification*
- ❏ Google - *Machine Learning Problem Framing*



# Ciência de Dados em *R*

**“No futuro, o pensamento estatístico será tão necessário para a cidadania eficiente como saber ler e escrever” *H.G.Wells***



## Principais Pacotes para AM em **R**

- `library(dplyr)`
- `library(Hmisc)`
- `library(e1071)`
- `library(caret)`
- `library(MASS)`
- `library(ggplot2)`
- `library(mlbench)`
- `library(rtree.part)`
- `library(ROCR)`
- `library(rpart)`





## Conjunto de Dados em *R*

- Obtendo amostra de dados:
  - `data(iris)`                      `data(Titanic)`
  - `data(economics)`              `data(mtcars)`
  - `data(diamonds)`                `data(Boston)`
- Fonte: [Exemplos de Conjunto de Dados](#)



# Análise Exploratória dos Dados em *R*

- `head(iris)`
- `nrow(Titanic)`
- `summary(economics)`
- `unique(mtcars)`
- `missing(diamonds)`
- `plot(Boston)`<sup>1</sup>

<sup>1</sup> Fonte: [R Documentation Plot Function](#)



# Pré-Processamento de Dados em *R*

- Conjuntos de dados:
  - Treinamento (***train***) e Teste (***test***)
    - `part<-sample_frac(data, 0.8)`
    - `treino<-as.numeric(rownames(part))`
    - `teste<data[-treino,]`



# Pré-Processamento de Dados em *R*

- Treinamento (*train*) e Teste (*test*)
  - `part<-sample(nrow(data), round(nrow(data)*0.8))`
  - `treino<-data[part, ]`
  - `test<data[-part,]`



# Pré-Processamento de Dados em *R*

- Conjuntos de dados:
  - Treinamento (***train***) e Teste (***test***)
    - **train**<-sample\_frac(data, 0.8)
    - treino<-as.numeric(rownames(**train**))
    - **test**<data[-treino,]



# Aplicar Modelo aos Dados em *R*

- Ajuste dos dados:
  - “Fitting” (***fit***)
    - **modelo** <- *model.fit*(data\$target, train)
- Prevendo com os dados:
  - “Predicting” (***predict***)
    - **resultado** <- model.predict(**modelo**, test)

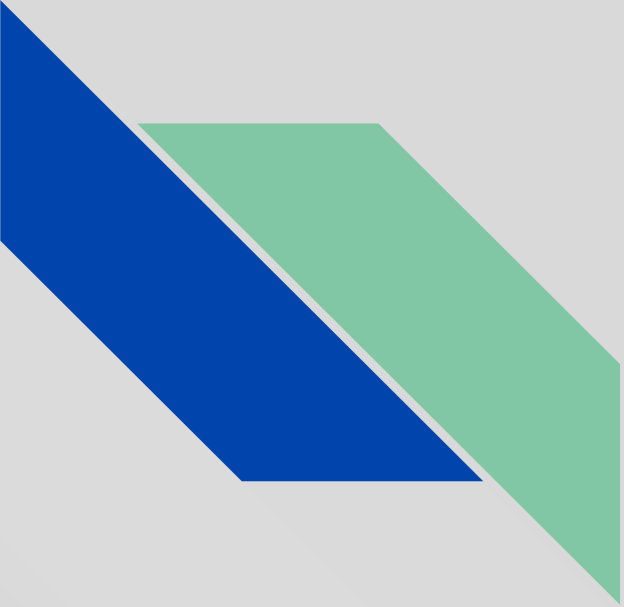


**Que tal uma  
parada?**









# Algoritmos

**"Não são tanto as coisas que não sabemos que nos metem em confusões. São as coisas que pensamos que sabemos." *Artemus Ward***



# Aprendizado de Máquina Supervisionado

- Aprendizado de máquina (AM) é o estudo científico de algoritmos e modelos estatísticos que os sistemas de computador usam para realizar uma tarefa específica sem usar instruções explícitas, confiando em padrões e inferência



# Algoritmos de Classificação

- Constróem um modelo matemático com base nos dados de amostra e seus rótulos, para fazer previsões ou conduzir decisões sem ser explicitamente programado para executar as tarefas



# Algoritmos de Classificação

- Naive Bayes
- Árvore de Decisão
- Regressão Logística
- Florestas Aleatórias
- Aprendizado baseado em instância
- Seleção de Atributos
- Máquina de Vetor de Suporte



# Naive Bayes

*Por Gustavo Alexandre*



# Naive Bayes

- É um algoritmo probabilístico simples baseado no teorema de Bayes
- Utiliza dados de treino para formar um modelo baseado na evidência das features nos dados
- O algoritmo supõe que há uma independência entre as features do modelo



## Naive Bayes em *R*

- `install.packages("e1071")`
- `library(e1071)`
- `modelo <- naiveBayes(alvo~., data=dados)`
- `result <- predict(modelo, dados)`
- `table(result, dados$alvo)` *#Matriz de Confusão*





# Árvore de Decisão

*Por Gustavo Alexandre*



# Árvore de Decisão

- É um modelo interpretável com gráfico no formato de árvore e demonstra visualmente as regras e probabilidades até os resultados
- Este algoritmo funciona tanto para problemas de **classificação** quanto para **regressão**

**Exemplo:** [RPubs | Árvore de Decisão por Fabrício Barth](#)



## Árvore de Decisão em *R*

- `install.packages("rpart")`
- `library(rpart)`
- `install.packages("rpart.plot")`
- `library(rpart.plot)`
- `fitDTree <- rpart(alvo~., treino)`
- `rpart.plot(fitDTree)` *#Gera a Árvore*



# Regressão Logística

*Por Gustavo Alexandre*



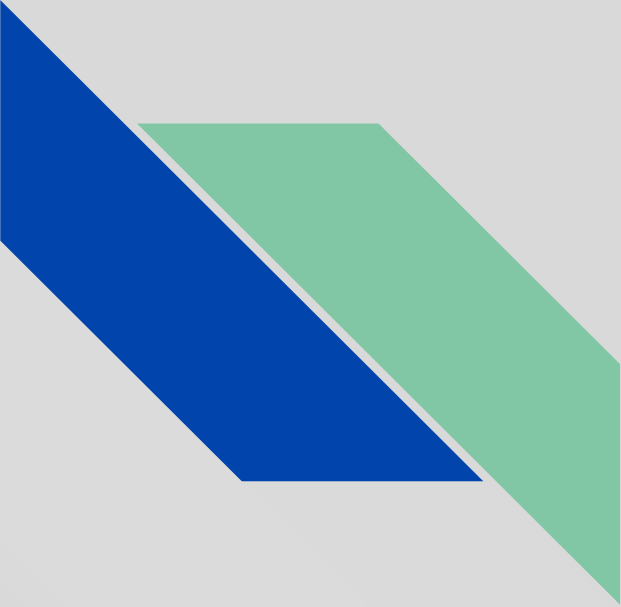
# Regressão Logística

- É um algoritmo estatístico muito utilizado para modelagem de saídas binárias
- Quando se quer medir a relação de uma variável dependente binária com uma ou mais variáveis independentes, é comum utilizar esta técnica



## Regressão Logística em *R*

- `library(stats)`
- `modelo <- glm(alvo~., data=treino, family="binomial")`
- `result <- predict(modelo, teste, type="response")`
- `summary(result)`



# **Florestas Aleatórias** ***(Random Forests)***

*Por Gustavo Alexandre*



## Florestas Aleatórias (*Random Forest*)

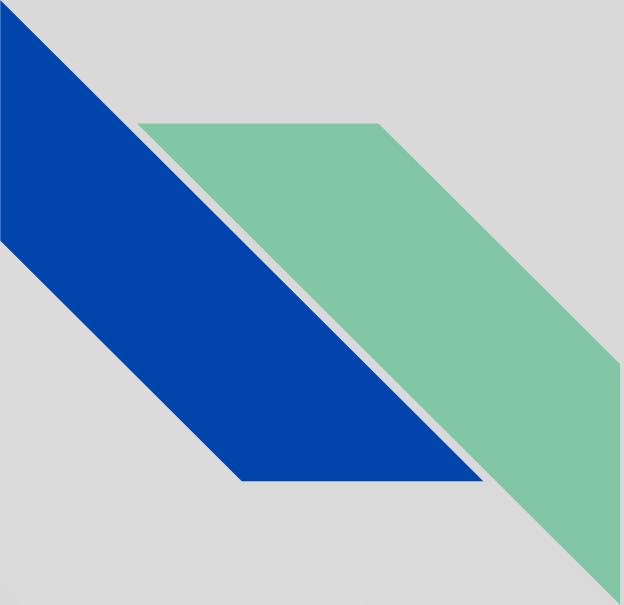
- É um modelo comum em classificação e regressão que permite construir uma multiplicidade de árvores de decisão durante o treinamento
- Resultando na melhor configuração dadas as árvores avaliadas





## Florestas Aleatórias (*Random Forest*) em *R*

- `install.packages("randomForest")`
- `library(randomForest)`
- `modelo <- randomForest(alvo~., data=dados, importance=TRUE, proximity=TRUE)`




# Aprendizado baseado em Instância (*KNN*)

*Por Gustavo Alexandre*



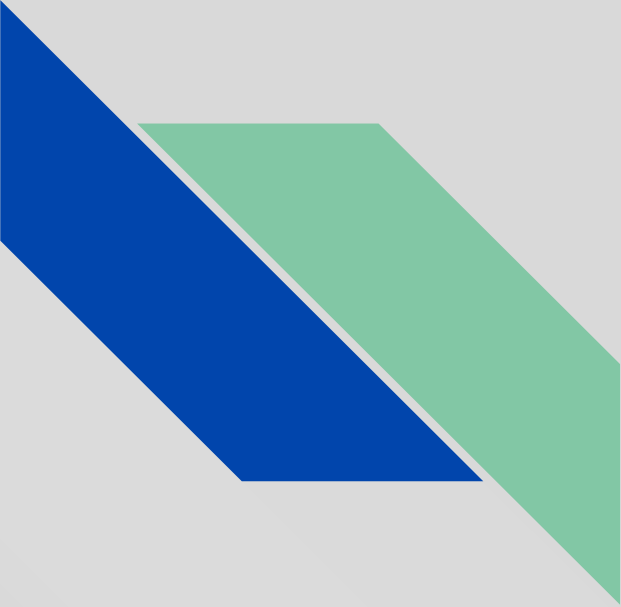
# Aprendizagem Baseada em Instâncias (*KNN*)

- Esse tipo de técnica armazena o conjunto de treinamento ao prever um valor (ou classe) para uma nova instância
- Utilizam-se de métricas de distância e similaridade durante o processo de treinamento
- Algoritmos: ***KNN***, máquinas de Kernel, redes RBF



# Aprendizagem Baseada em Instâncias (*KNN*) em *R*

- `install.packages("caret")`
- `library(caret)`
- `modelo <- trainControl(method="cv", number=5)`
- `fitKNN <- train(alvo~., data=dados, method= "knn",  
metric="Accuracy", trControl=modelo)`



# **Seleção de Atributos** ***(Feature Selection)***

*Por Gustavo Alexandre*



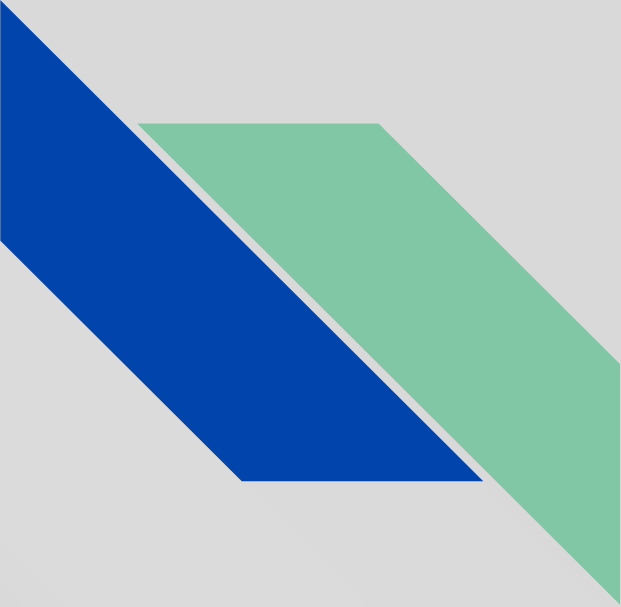
# Seleção de Atributos

- Aplicar **seleção de atributos** nos dados que podem conter atributos redundantes ou irrelevantes
- A remoção desse atributos não deve promover perdas significativas



## Seleção de Atributos em *R*

- `install.packages("caret")`
- `library(caret)`
- `modelo <- trainControl(method="cv", number=10)`
- `modelFS <- train(alvo~., data=dados, method="lvq",  
preProcess="scale", trControl=modelo)`
- `selecAtributos <- varImp(modelFS, scale=FALSE)`



# Máquina de Vetor de Suporte (SVM)

*Por Gustavo Alexandre*

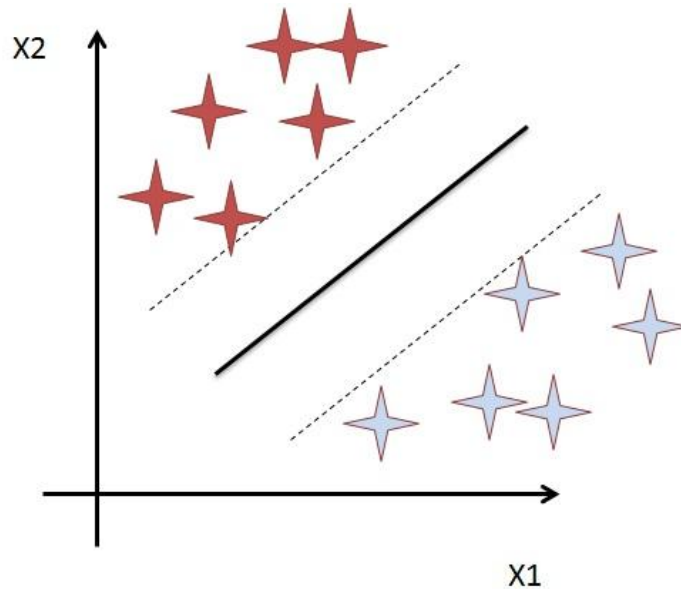




# Máquina de Vetor de Suporte (SVM)

- O SVM é um classificador linear binário não probabilístico
- SVM busca um *hiperplano* entre os dados a serem classificados e visa maximizar a distância entre os pontos, separando cada uma das classes

# Máquina de Vetor de Suporte (SVM)

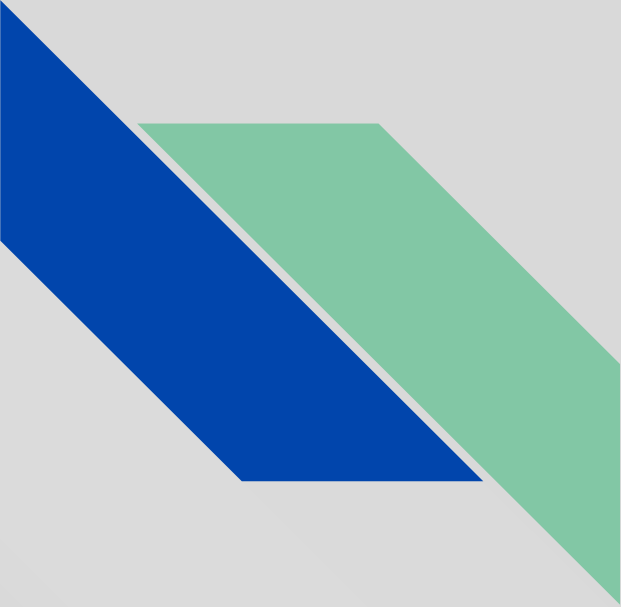




## Máquina de Vetor de Suporte (SVM) em **R**

- `install.packages("e1071")`
- `library(e1071)`
- `modelo <- svm(formula=alvo~., data=treino, type='C-classification', kernel='linear')`
- `resultado = predict(modelo, newdata=test)`

**Fonte:** [R Documentation | SVM Function](#)



# Avaliação

**"No futuro, o pensamento estatístico será tão necessário para a cidadania eficiente como saber ler e escrever" *H.G.Wells***



# Avaliação de Modelo

- A avaliação dos classificadores binários compara dois métodos de atribuição de um atributo binário, um dos quais é geralmente um método padrão e o outro está sendo investigado



# Avaliação de Modelo

- Existem muitas métricas que podem ser usadas para medir o desempenho de um classificador ou preditor e áreas diferentes têm preferências diferentes para métricas específicas devido aos seus objetivos



# Avaliação de Modelo

- **Exemplo de preferências:**
  - Na **medicina**, sensibilidade e especificidade são frequentemente usadas
  - Na **ciência da computação**, a precisão e a recordação são preferidas



# Definições

- **TP** = Verdadeiro Positivo (***True Positive***)
- **TN** = Verdadeiro Negativo (***True Negative***)
- **FP** = Falso Positivo (***False Positive***)
- **FN** = Falso Negativo (***False Negative***)





## **Métricas de Avaliação (Classificação)**

- **Acurácia**
- **Medida F1 (*F-Score*)**
- **Matriz de Confusão**
- **Curva ROC**



# Acurácia

- Segundo a Física, é a exatidão de uma medição ou de um instrumento de medição
- Para o AM, é uma métrica para avaliar modelos de classificação



# Acurácia

- Pode-se dizer, informalmente, que acurácia é a fração de predições que nosso modelo acertou.
- Formalmente, a acurácia tem a seguinte definição:

$$\text{Acuracia} = (TP + TN)/(TP+TN+FP+FN)$$



## Acurácia em *R*

- `devtools::install_github("selva/InformationValue")`
- `library(InformationValue)`
- `result <- predict(modelo, newdata=teste, type="response")`
- `y_pred <- ifelse(result>0.5, True, False)`
- `accuracy <- mean(y_pred == dados$alvo)`



## Medida F1 (*F-Score*)

- É a métrica com melhor aceitação dentre a avaliação de um teste de classificação
- Tem sido amplamente utilizada em NLP no Reconhecimento de Entidade Nomeada
  - ***Specificity*** =  $TP / (TP + FP)$
  - ***Sensibility*** =  $TN / (TN + FN)$



## Medida F1 (*F-Score*)

- É uma média harmônica (***Specificity*** e ***Sensibility***) em que a pontuação atinge seu melhor valor em **1** e a pior em **0**.
- Formalmente, **F1** tem a seguinte definição:

$$\text{F-Score} = \frac{2 \times (\text{Specificity} \times \text{Sensibility})}{(\text{specificity} + \text{sensibility})}$$



## Medida F1 (F-Score) em *R*

- `library(InformationValue)`
- `result <- predict(modelo, newdata=teste, type="response")`
- `y_pred <- ifelse(result>0.5, True, False)`
- `InformationValue::fscore(dados$alvo, y_pred)`



# Matriz de Confusão

- É também conhecida como "**Matriz de Erro**", a qual permite a visualizar o desempenho em **Aprendizado Supervisionado** (classificação)
- Em **Aprendizado Não-Supervisionado**, é geralmente chamada de "**Matriz de Correspondência**", com duas dimensões e classes em ambas as dimensões





# Matriz de Confusão

		Valor Previsto	
		Positivo	Negativo
Valor Verdadeiro	Negativo	Verdadeiros Positivos	Falsos Negativos
	Positivo	Falsos Positivos	Verdadeiros Negativos



## Matriz de Confusão em *R*

- ***Dica***: Use o Modelo gerado na **Regressão Logística**
- **Exemplo 1** (*Matriz de Confusão somente*):
  - `matriz <- ifelse(result > 0.5, "True", "False")`
  - `table(matriz, teste$alvo)`
- **Exemplo 2** (*Matriz de Confusão e outras métricas*):
  - `confusionMatrix(factor(matriz), teste$alvo)`



## Curva ROC

- É uma representação gráfica que ilustra o desempenho de um classificador binário e como o seu limiar de discriminação é variado
- A curva ROC foi desenvolvida na Segunda Guerra Mundial para detecção de objetos inimigos nas batalhas

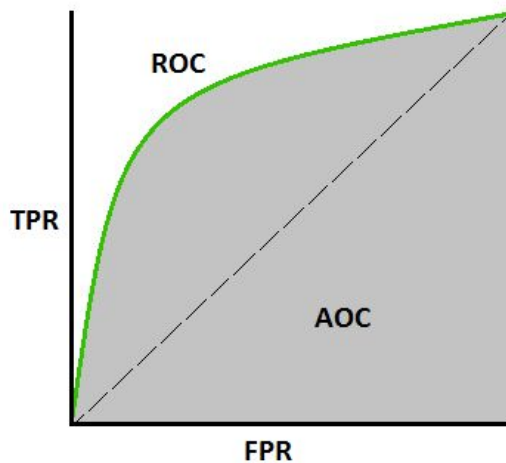


## Curva ROC

- Atualmente auxilia na psicologia, medicina, radiologia, aprendizado de máquina, mineração de dados, entre outros
- É obtida pela representação da fração de **Positivos Verdadeiros dos Positivos Totais** ( $TPR = PV/P$ ) versus a fração de **Positivos Falsos dos Negativos Totais** ( $FPR = PF/N$ ), em várias configurações do limite

# Curva ROC

- Graficamente, a Curva ROC é expressa da seguinte forma:





## Curva ROC em *R*

- `install.packages("ROCR")`
- `library(ROCR)`
- `pred <- prediction(dados, dados$alvo)`
- `perf <- performance(pred, "tpr", "fpr")`
- `plot(perf)`

**Fonte:** [R Documentation | ROCR Prediction](#)



## Curva ROC em *R*

- `install.packages("caTools")`
- `library(caTools)`
- 
- `caTools::colAUC(p, teste["alvo"], plotROC=True)`

Fonte: [Seção 5.4.2 | ROC Curve](#)

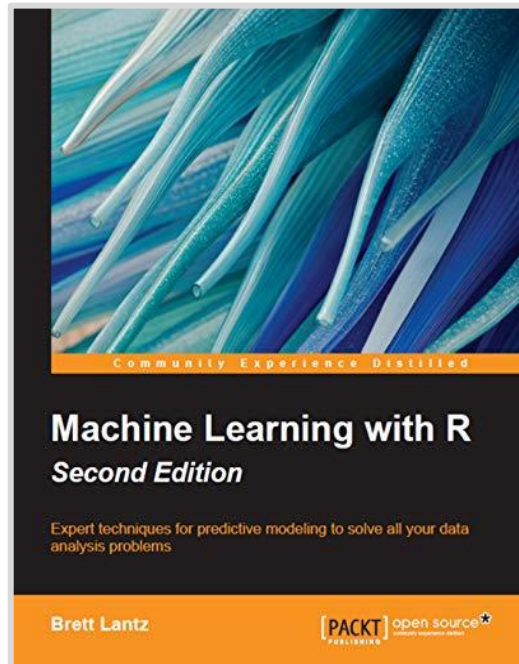
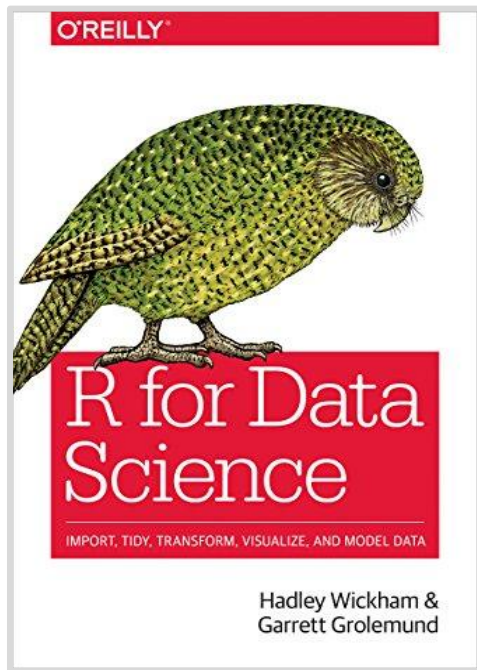


# Leitura Complementar

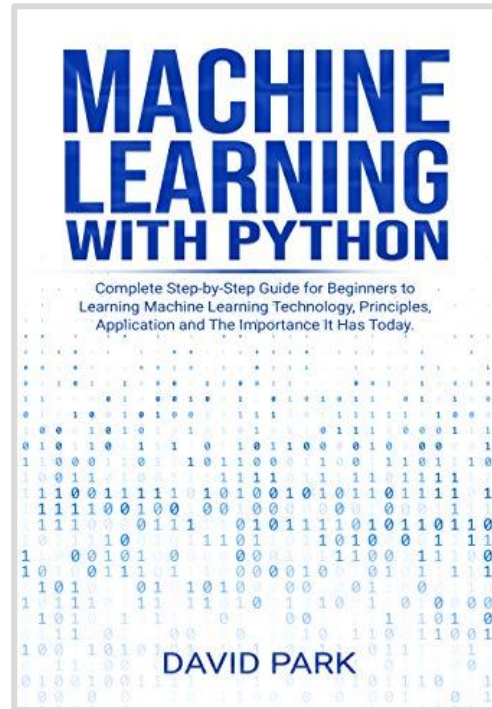
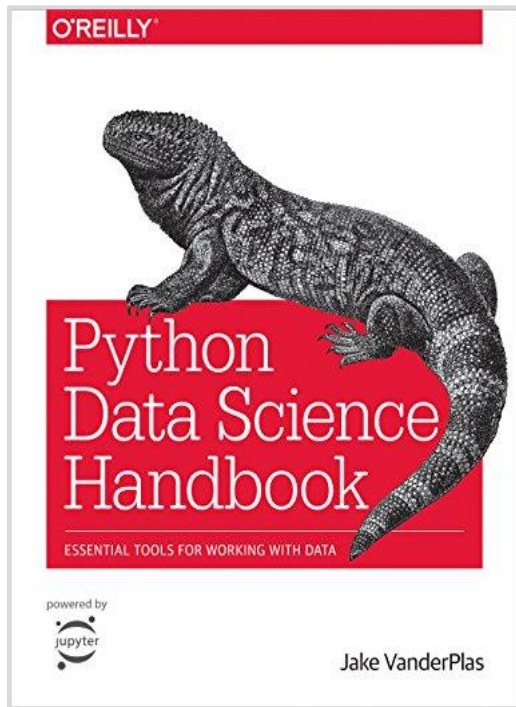
- ❑ KOHAVER, Ron; PROVOST, Foster. *Glossary of terms. Machine Learning.*
- ❑ Coursera - *Evaluation Metrics in Classification (IBM)*
- ❑ Blog - *Machine Learning Plus | Evaluation Metrics*



# Bibliografia Recomendada (em **R**)



# Bibliografia Recomendada (em *Python*)







**Muito Obrigado!**





## Referências Bibliográficas

- ❑ Lantz, B. (2013). **Machine learning with R.** Packt Publishing Ltd.
- ❑ CONWAY, Drew; WHITE, John. (2012). **Machine learning for hackers.** O'Reilly Media, Inc.
- ❑ ZUMEL, Nina; MOUNT, John. (2014). **Practical data science with R.**