

Aprendizado de máquina

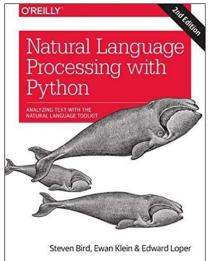
Fundamentos e aplicações em processamento de linguagem natural

Felipe Navarro Balbino Alves

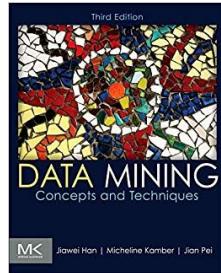
Github do curso

https://github.com/fnbalves/curso_machine_learning/

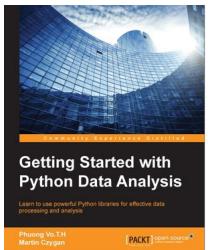
Bibliografia de interesse



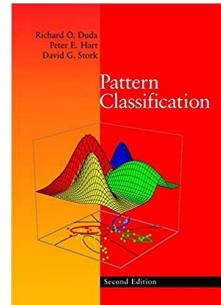
Natural Language Processing with Python
Steven Bird (Disponível online)



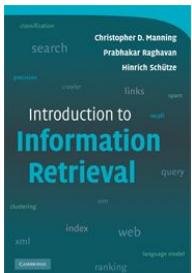
Data Mining - Concepts and Techniques
Han & Kamber



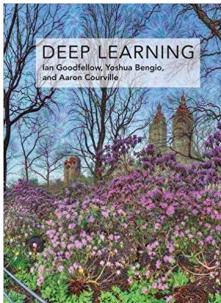
Getting started with Python Data Analysis
Phuong Vo. T.H



Pattern Classification
Han & Kamber



Introduction to information Retrieval
Cristopher D. Manning



Deep Learning
Ian Goodfellow (Disponível online)

Bibliografia de interesse - Material online

NLTK book:

<https://www.nltk.org/book/>

Introduction to Information Retrieval:

<https://nlp.stanford.edu/IR-book/>

Scikit-learn documentation:

<https://scikit-learn.org/stable/>

The Deep Learning book:

<https://www.deeplearningbook.org/>



Classification

Identifying to which category an object belongs to.
Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ...
[— Examples](#)

Regression

Predicting a continuous-valued attribute associated with an object.
Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ...
[— Examples](#)

Clustering

Automatic grouping of similar objects into sets.
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ...
[— Examples](#)

Dimensionality reduction

Reducing the number of random variables to consider.
Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization.
[— Examples](#)

Model selection

Comparing, validating and choosing parameters and models.
Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics.
[— Examples](#)

Preprocessing

Feature extraction and normalization.
Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction.
[— Examples](#)

O que é aprendizado de máquina?

Exemplo prático: precificar um imóvel

Seu trabalho é ajudar um corretor de imóveis a estimar o preço de uma propriedade.

Como podemos atacar o problema?

Valor Venda	R\$ 290.000
Quartos	
	2
Banheiros	
	2
M² total	
	120
Vagas	
	1

O que é aprendizado de máquina?

Exemplo prático: precificar um imóvel

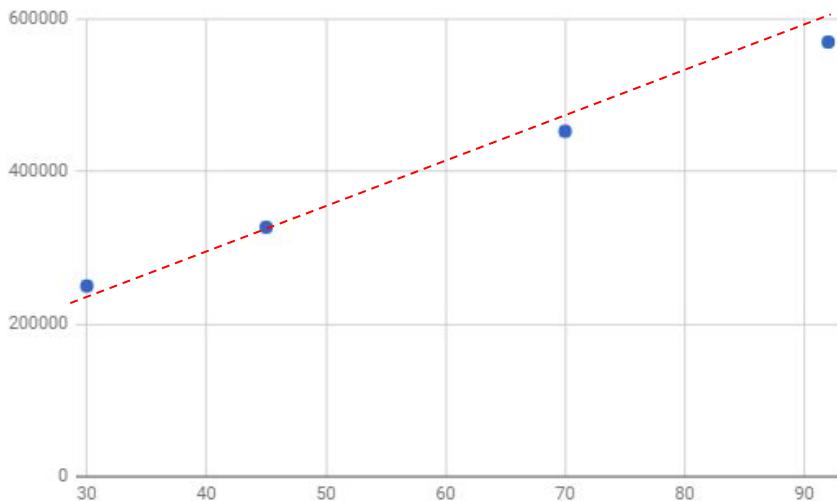
O corretor lhe fornece a seguinte planilha de dados de precificação anteriores

Imóvel	Tamanho do imóvel em metros quadrados	Preço do imóvel em reais
Rua da Palma, 467	30	250.000
Conde de irajá 344	45	325.100
Agamenon magalhães 56	70	453.000
Joaquim Nabuco 443	92	570.000

O que é aprendizado de máquina?

Exemplo prático: precificar um imóvel

Você decide plotar os dados



Você descobre que a relação é aproximadamente uma reta, de equação

$$100000 + 5000 \times (\text{tamanho do imóvel})$$

O que é aprendizado de máquina?

Exemplo prático: precificar um imóvel

Agora vamos supor que a planilha fornecida fosse assim:

Imóvel	Tamanho do imóvel (m ²)	Número de quartos	Tem gás encanado	Impostos em dia?	Vagas de garagem	Cidade	Construtora
Imóvel 1	30	1	Sim	Não	1	Recife	(dado faltante)
Imóvel 2	45	3	Não	Sim	2	Jaboatão	Construtora legal
Imóvel 3	70	2	Não	(dado faltante)	2	Paulista	A sua construtora
Imóvel 4	92	5	Sim	Não	3	Olinda	A melhor construtora

...

O que é aprendizado de máquina?

Exemplo prático: precificar um imóvel

E agora, como fazemos para **extrair uma regra** de forma visual?

Imóvel	Tamanho do imóvel (m ²)	Número de quartos	Tem gás encanado	Impostos em dia?	Vagas de garagem	Cidade	Construtora
Imóvel 1	30	1	Sim	Não	1	Recife	(dado faltante)
Imóvel 2	45	3	Não	Sim	2	Jaboatão	Construtora legal
Imóvel 3	70	2	Não	(dado faltante)	2	Paulista	A sua construtora
Imóvel 4	92	5	Sim	Não	3	Olinda	A melhor construtora

...

O que é aprendizado de máquina?

Aprendizado de máquina é um sub-ramo da ciência da computação especializado no reconhecimento automático de **padrões** a partir de **dados**

Inteligência artificial x aprendizado de máquina

Inteligência artificial é um conceito mais amplo e trata de máquinas capazes de realizar tarefas consideradas “inteligentes”. Abrange temas como Teoria dos jogos, Sistemas de busca, representação de conhecimento, planejamento, entre outros

Inteligência artificial x aprendizado de máquina

● machine learning
Termo de pesquisa

● artificial intelligence
Termo de pesquisa

+ Adicionar comparação

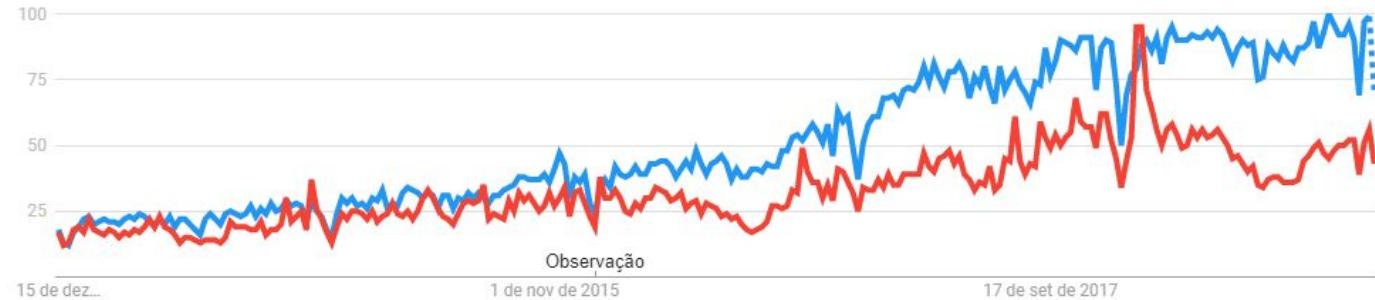
Estados Unidos ▾

Nos últimos 5 anos ▾

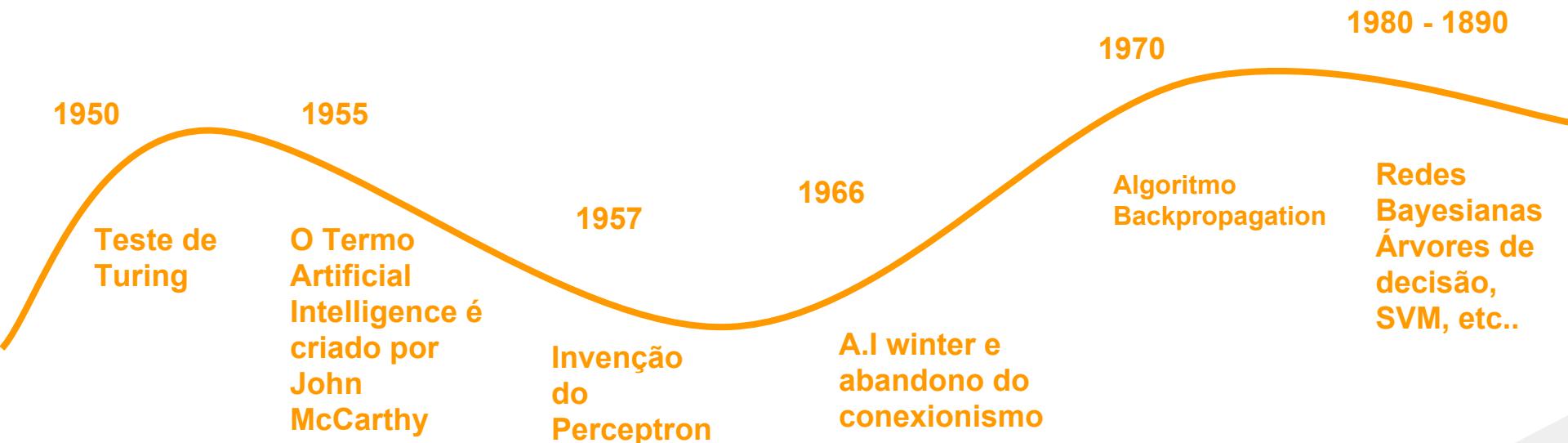
Todas as categorias ▾

Pesquisa na Web ▾

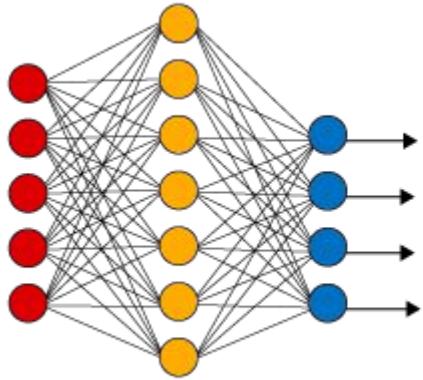
Interesse ao longo do tempo 



Evolução do AM

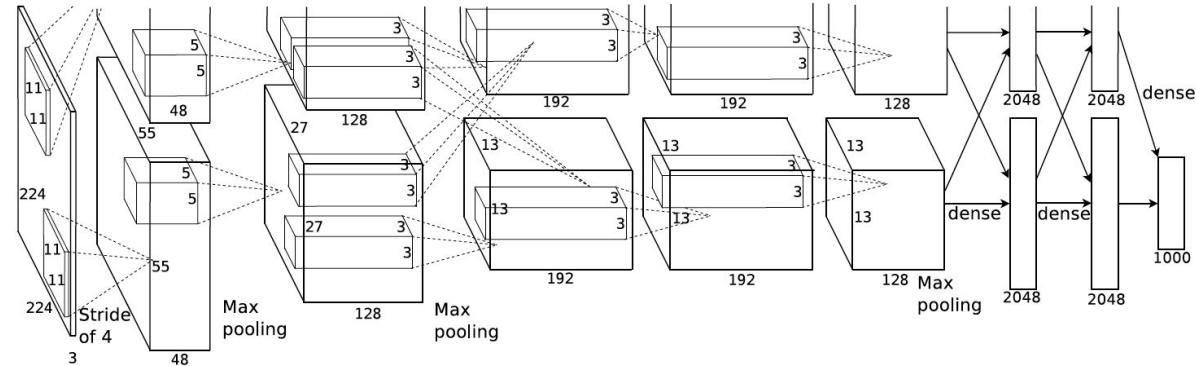


Surgimento do deep learning

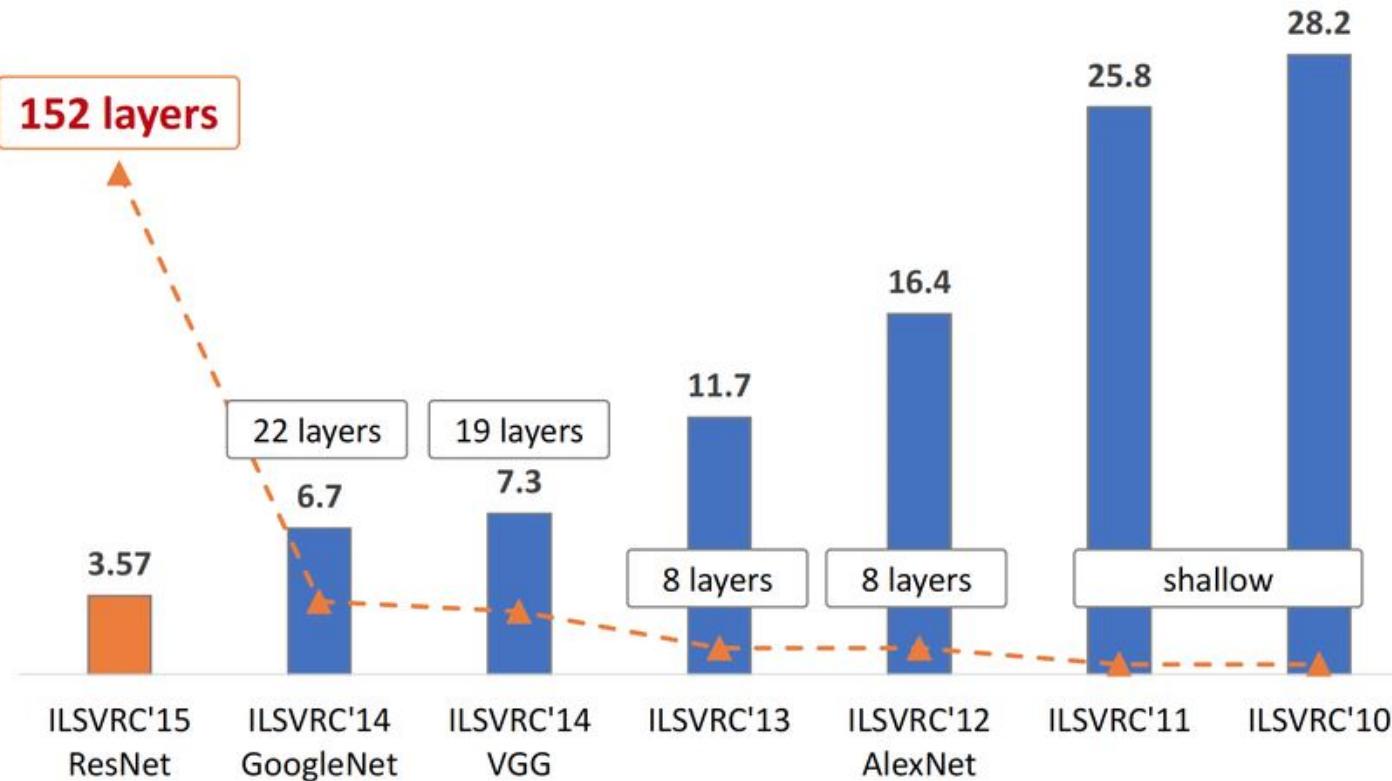


Shallow network

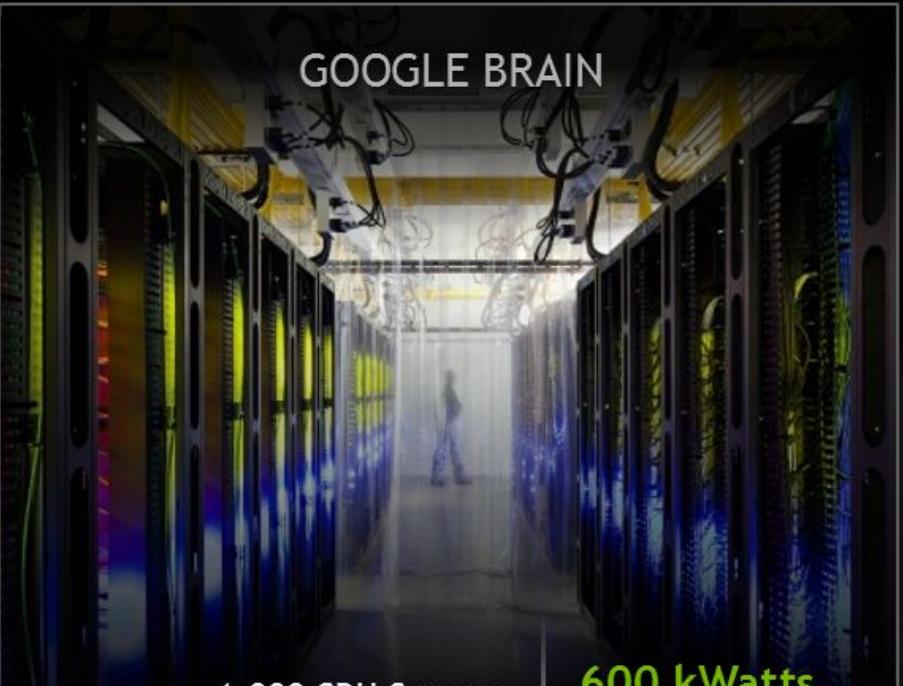
Vs Alex net



Surgimento do deep learning



Evolução de poder computacional



GOOGLE BRAIN

1,000 CPU Servers
2,000 CPUs • 16,000 cores

600 kWatts
\$5,000,000

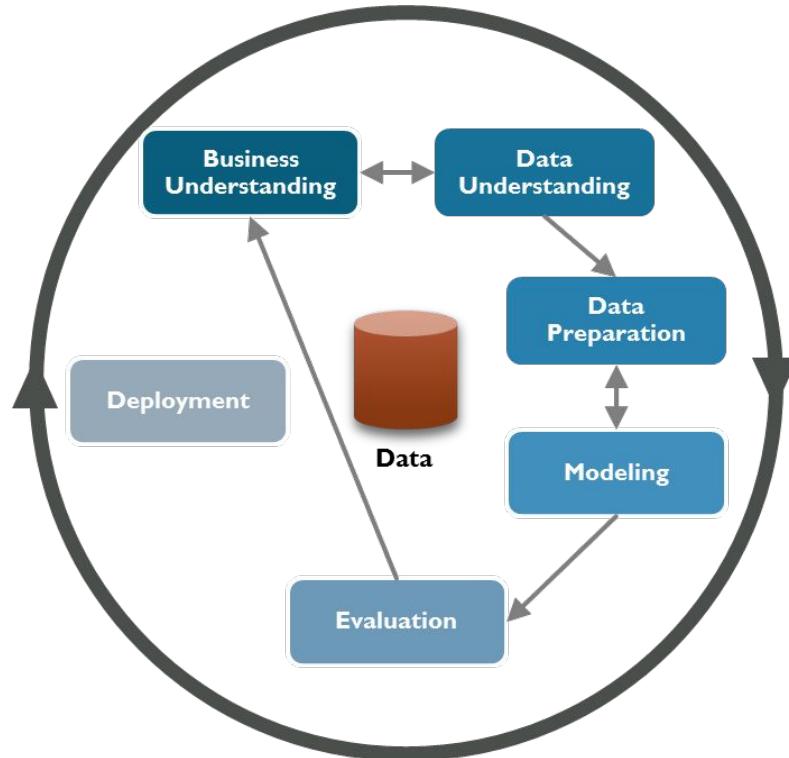
300X energy efficiency
400X lower cost
Fits under a desk



1 Titan Z-Accelerated Server
3 Titan Zs • 17,280 cores

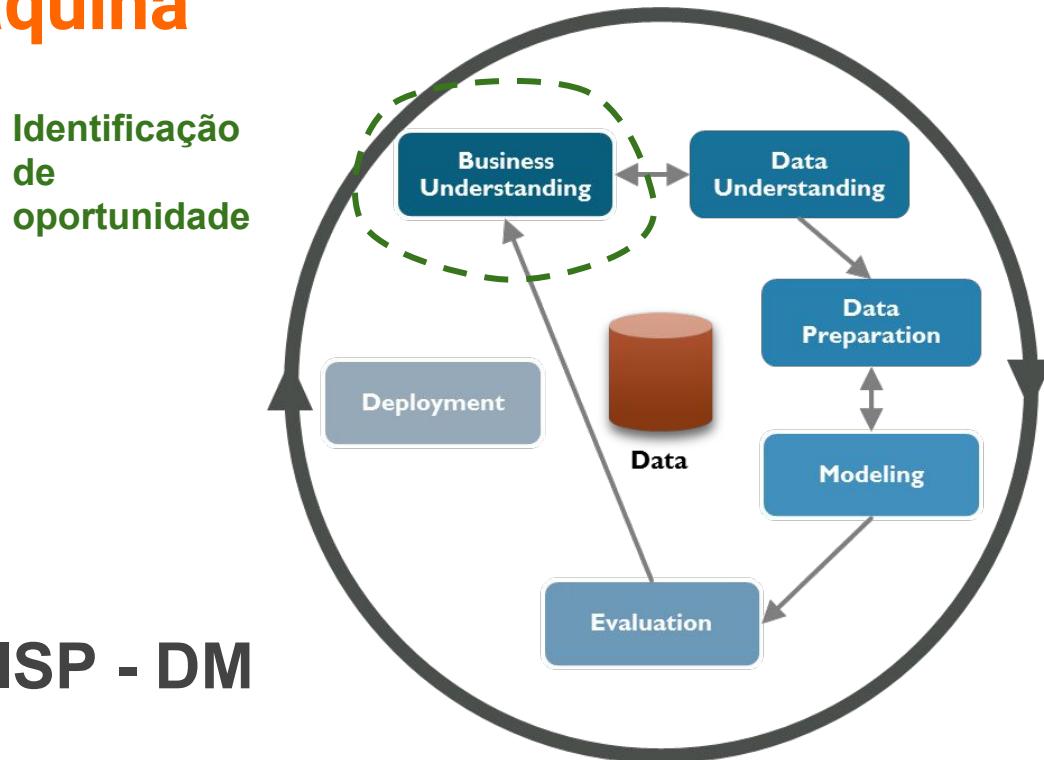
2 kWatts
\$12,000

Como funciona um projeto de Aprendizado de máquina



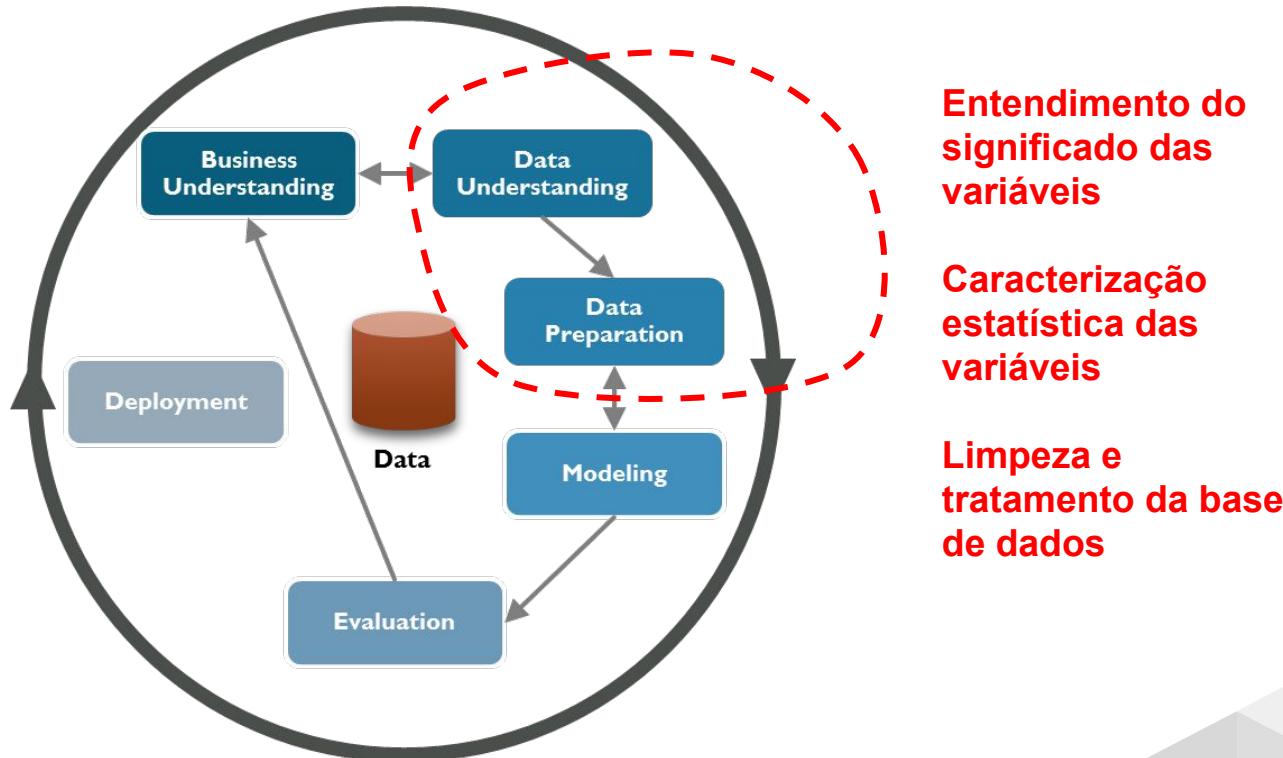
CRISP - DM

Como funciona um projeto de Aprendizado de máquina



CRISP - DM

Como funciona um projeto de Aprendizado de máquina



CRISP - DM

Em relação aos dados

O primeiro passo para a correta utilização de uma variável é entender o que ela representa

Dessa forma, evitamos utilizar **dados a posteriori**

Exemplo: Suponha que se deseja criar um modelo para prever a complexidade do conserto de uma máquina a partir do log de eventos da mesma. Podemos usar como feature a **peça que foi substituída?**

Caracterização estatística dos dados

A seguir, precisamos identificar qual o tipo da variável em questão:

Variável numérica: Tamanho do terreno (30m², 49 m²,)

Variável categórica: IPTU em dia? (Sim ou não)

Dentro das variáveis categóricas, podemos classificá-las em:

Variável nominal: Não existe uma ordem de grandeza. Ex: sexo, estado civil

Variável ordinal: Existe uma ordem entre as categorias. Ex: escolaridade

Uma variável categórica pode ser nominal ou ordinal **dependendo do contexto.**

...

Caracterização estatística dos dados

Para **variáveis numéricas**, é importante identificar durante o entendimento dos dados, as seguintes grandezas:

Porcentagem de Missing data

Média

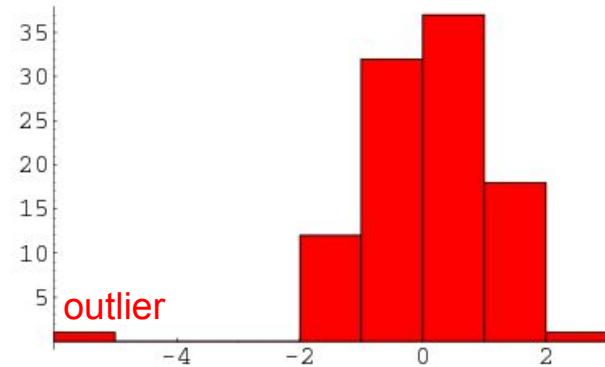
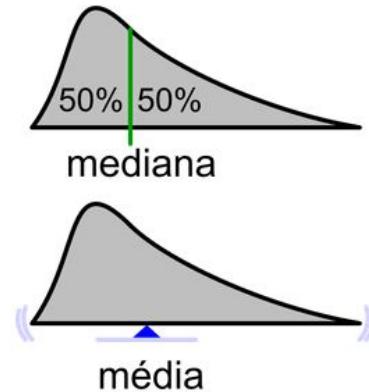
$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i$$

Variância

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Outliers

Mediana



Histograma

Caracterização estatística dos dados

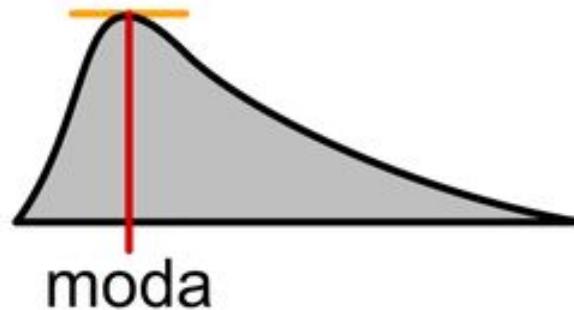
Para **variáveis categóricas**, é importante identificar durante o entendimento dos dados, as seguintes grandezas:

Porcentagem de Missing data

Moda

Outliers

Histograma



Análise de correlação

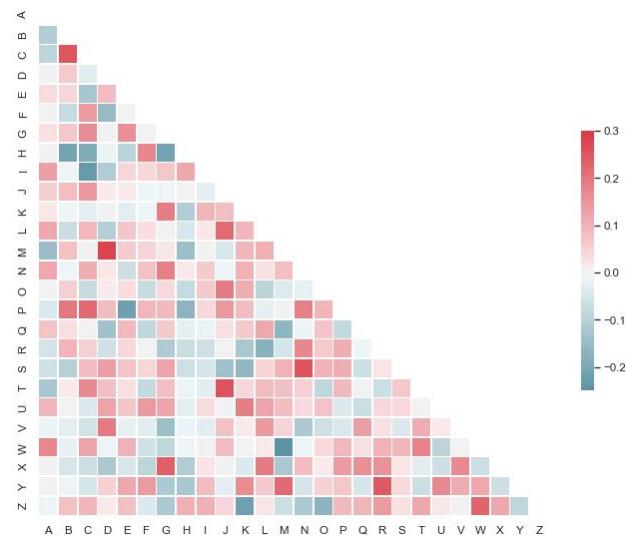
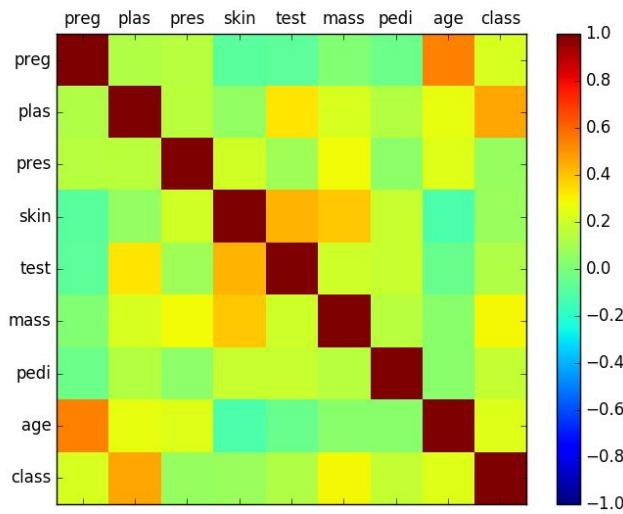
Muitas vezes, uma mesma informação é representada de múltiplas formas em uma mesma base de dados. Variáveis redundantes são um **problema** para muitos algoritmos de aprendizado. Desta forma, se faz necessária uma análise de correlação

$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X) \cdot \text{var}(Y)}}$$

Se $\rho > 0$, as variáveis tendem a crescer ou decrescer ao mesmo tempo. Se $\rho < 0$, as variáveis tendem a ter comportamento oposto.

Análise de correlação

Nas bibliotecas que mostraremos ao longo do treinamento, a análise de correlação pode ser feita a partir de uma **matriz de correlação**



Missing data

O tratamento de missing data varia bastante de acordo com o contexto e com o negócio. É necessário tentar entender qual o **motivo** do aparecimento do valor faltante. Para variáveis numéricas, temos algumas opções:

1. **Listwise deletion:** Eliminar registros com o missing data;
2. **Average imputation:** Substituir um dado faltante pela média dos demais;
3. **Regression substitution:** Criação de um modelo para prever o missing value.

Missing data

No caso de variáveis categóricas, as opções são um pouco diferentes:

1. **Listwise deletion:** Eliminar registros com o missing data;
2. **Average imputation:** Substituir um dado faltante pela **moda** dos demais;
3. **Classification substitution:** Criação de um modelo para prever o missing value;
4. Criação de uma categoria extra para identificar missing values;

Engenharia de atributos

Suponha que se deseje criar um modelo para calcular a probabilidade de um indivíduo ter uma **doença vascular**.

Suponha que você possua dois parâmetros: peso (kg) e altura (m).

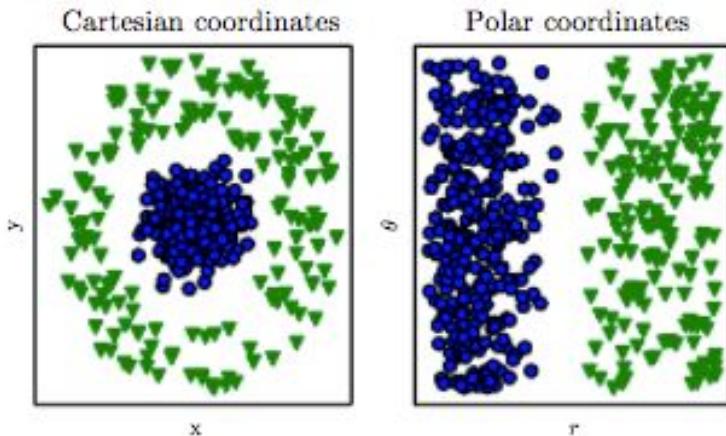
Provavelmente, estas duas variáveis em conjunto não serão mais relevantes do que a variável combinada:

$$\text{imc} = \text{peso} / \text{altura} ^ 2$$

Engenharia de atributos

Para criar novas variáveis de **grande relevância**, é, em geral necessário um grande conhecimento específico do domínio.

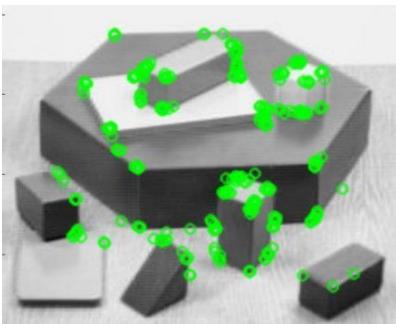
Exemplo 2:



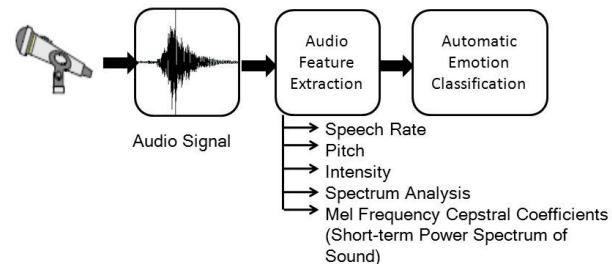
Engenharia de atributos

A engenharia de features é utilizada em todas as modalidades do aprendizado de máquina. Geralmente exigem **conversas com especialistas**

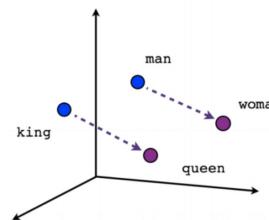
Processamento de imagens



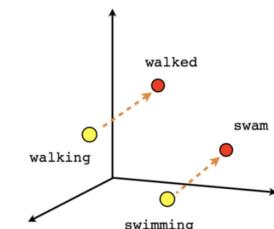
Audio Feature Extraction



Processamento de texto



Male-Female



Verb tense

Mudança de granularidade

Com frequência o processo de análise de dados exige a mudança da granularidade da informação. Como exemplo, suponha que temos à disposição dados de **alunos** do ensino médio, mas estamos interessados em analisar **instituições** de ensino médio.

Como devemos lidar com variáveis como a idade dos alunos, a escolaridade dos pais, etnia, etc... quando analisamos a instituição como um todo?

Mudança de granularidade - variáveis numéricas

Para variáveis numéricas, podemos criar novas variáveis que representam características estatísticas dos dados

Média

Desvio padrão

Max

Min

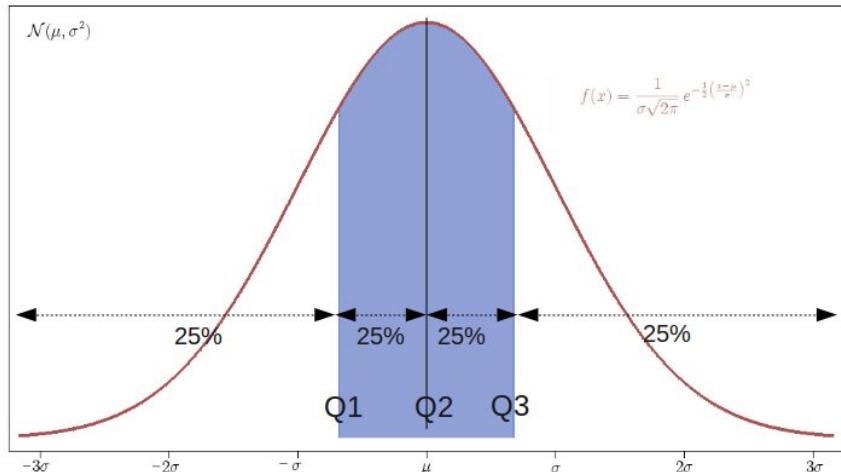
Mediana

Moda

Quantis

Quantis

Quantis são valores que dividem a área sob a função distribuição de probabilidade em partes iguais

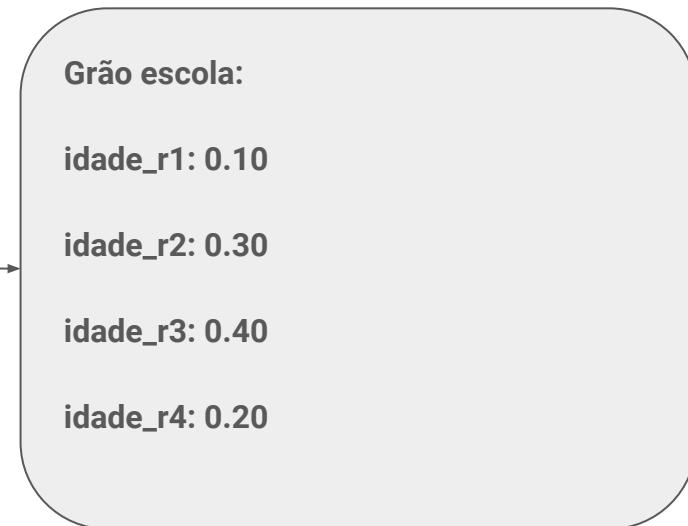
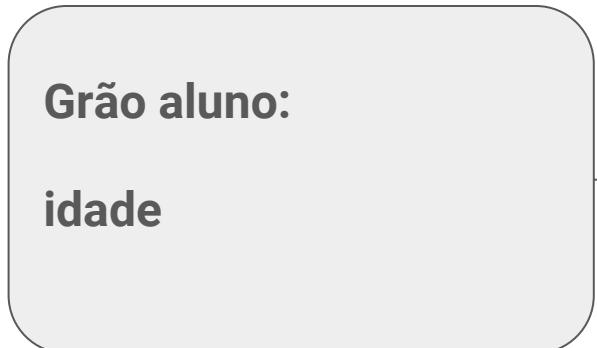


No grão maior, podemos colocar a quantidade de elementos que apareceu em cada uma das regiões

Quantis

Quantis são valores que dividem a área sob a função distribuição de probabilidade em partes iguais

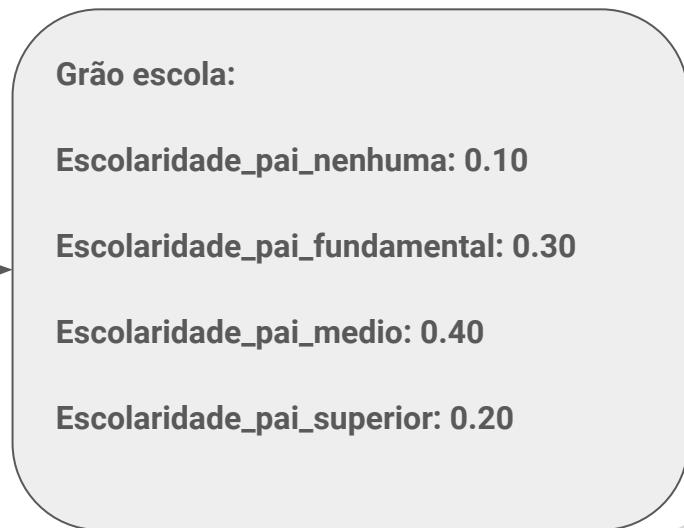
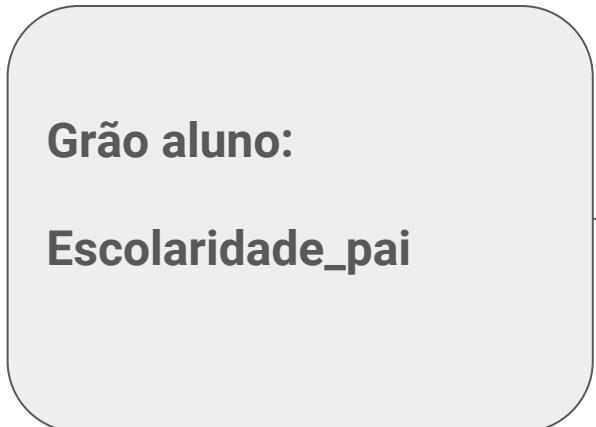
Ex:



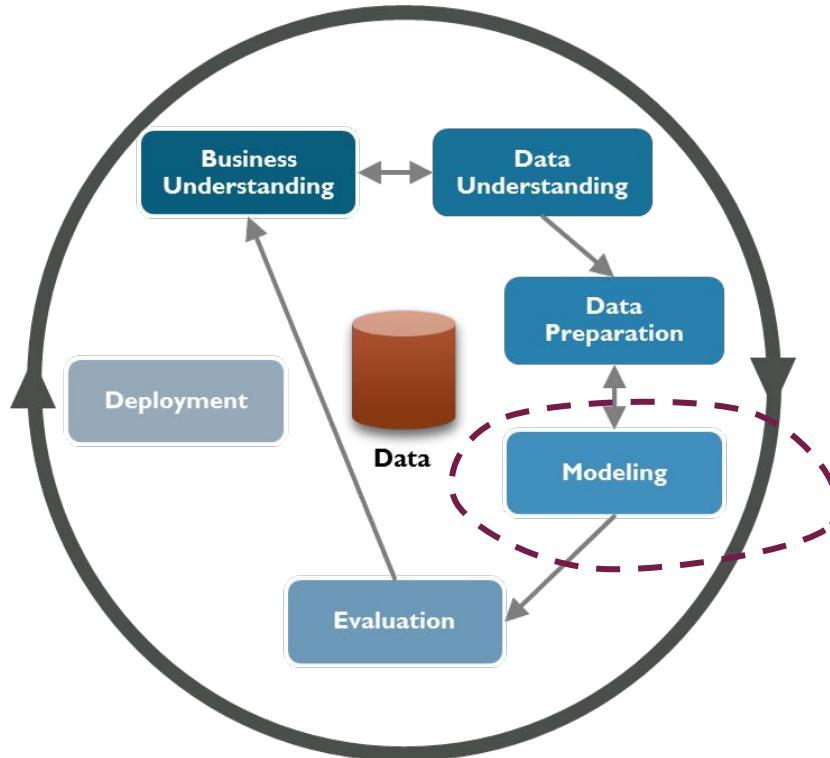
Mudança de granularidade - variáveis categóricas

Para variáveis categóricas, podemos fazer algo similar à estratégia dos quantis, inserindo quantos elementos apareceram em cada classe, ou até mesmo inserindo a frequência relativa

Ex:



Como funciona um projeto de Aprendizado de máquina



Neste momento,
tentaremos
identificar o padrão
nos dados

CRISP - DM

Tipos de modelos

Podemos dividir os modelos de aprendizado de máquina em duas grandes classes: modelos supervisionados e modelos não supervisionados.

O exemplo de precificação de um imóvel é uma forma de **aprendizado supervisionado**, onde existe um “professor” que diz a resposta correta para vários exemplos de entrada. Dizemos que temos **dados rotulados**.

Existe o **aprendizado não supervisionado**, onde não existem dados rotulados, o objetivo é encontrar **grupos de similaridade**

Aprendizado supervisionado

Dentro do aprendizado supervisionado, podemos ter dois tipos de problemas

Régressão: Exemplo do preço da casa

Classificação: Classificar uma entrada em um número discreto de possibilidades
(Ex: reconhecer se uma imagem é um gato ou cachorro)

Ganho de informação

No caso de algoritmos de classificação, existe outra métrica que pode ser utilizada para analisar os dados, que é o **Ganho de informação**, que se baseia na medida **entropia**

$$\text{Entropia}(S) = - \sum p_i \log_2 p_i$$

Dois casos:

classe 1:
probabilidade (50%)

classe 2:
probabilidade (50%)

$$\text{Entropia} = -0.5 \cdot \log_2(0.5) + -0.5 \cdot \log_2(0.5) = 1$$

classe 1:
probabilidade (90%)

classe 2:
probabilidade (10%)

$$\text{Entropia} = -0.9 \cdot \log_2(0.9) + -0.1 \cdot \log_2(0.1) = 0.46$$

Ganho de informação

O Ganho de informação de uma variável é definida da seguinte forma:

$$IG(T, a) = H(T) - H(T|a)$$

$$S_a(v) = \{\mathbf{x} \in T | x_a = v\}$$

$$H(T|a) = \sum_{v \in vals(a)} \frac{|S_a(v)|}{|T|} \cdot H(S_a(v))$$

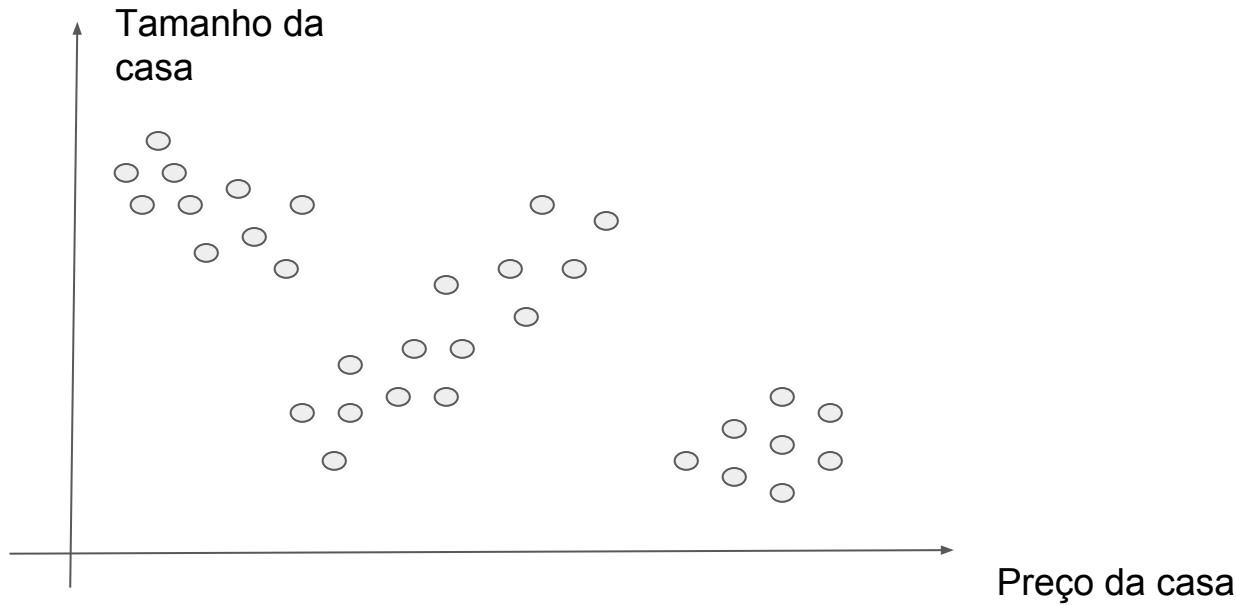
Variáveis com maior ganho de informação tem maior chance de serem relevantes no processo de classificação

Aprendizado não supervisionado

No **aprendizado não supervisionado**, onde não existem dados rotulados, o objetivo é encontrar **grupos de similaridade**

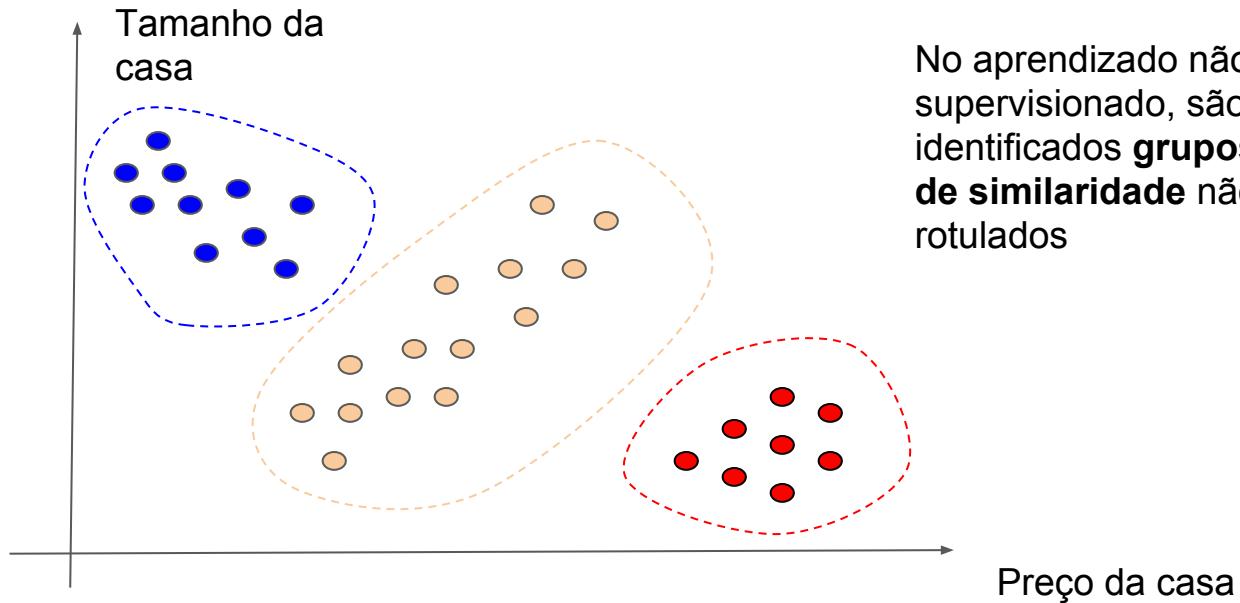
Aprendizado não supervisionado

Exemplo



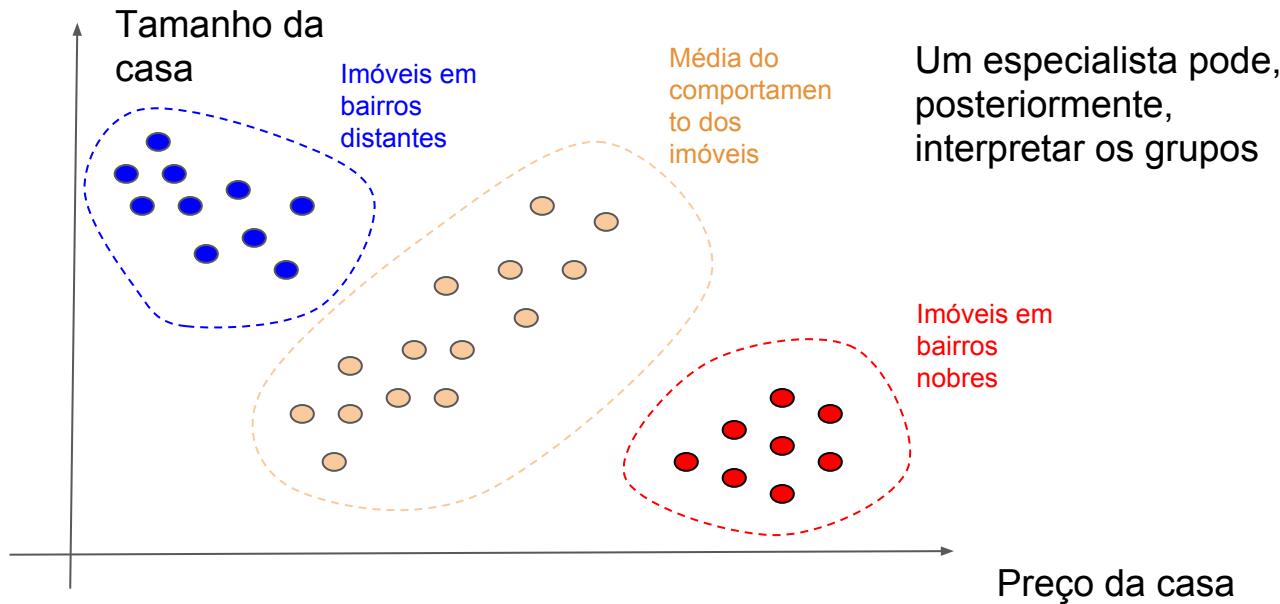
Formas de aprendizado

Exemplo



Formas de aprendizado

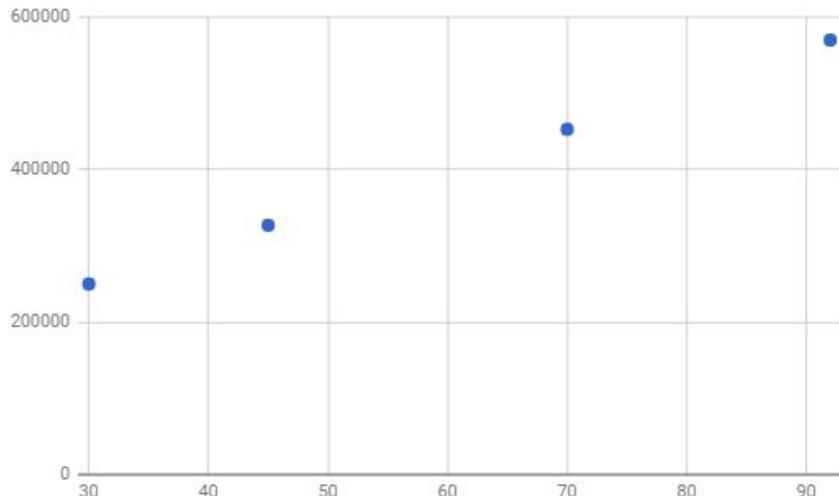
Exemplo



Analisando o padrão encontrado

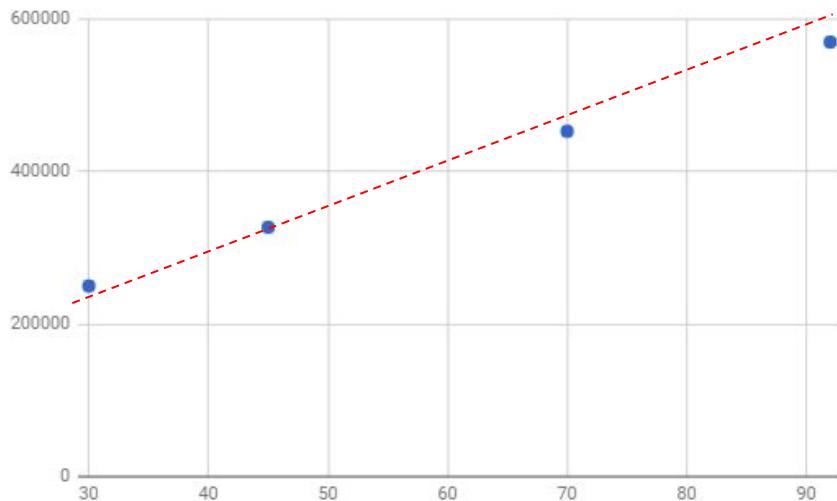
Exemplo prático: precificar um imóvel

Voltando ao exemplo da precificação do imóvel. Temos os dados. Qual é o **padrão**?



Avaliando o padrão encontrado

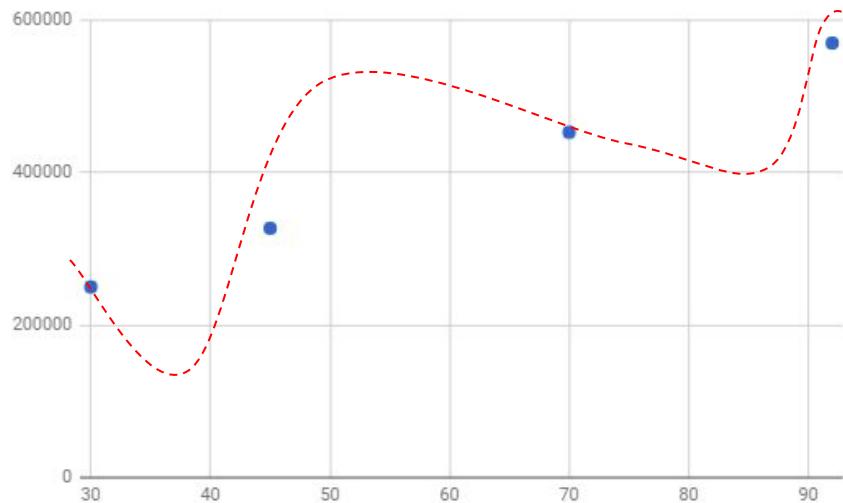
No âmbito da ciência de dados, o padrão também é chamado de **hipótese**



Neste caso, a nossa hipótese é que a lei que regia o fenômeno era uma linha reta, mas **precisava ser?**

Qual o padrão?

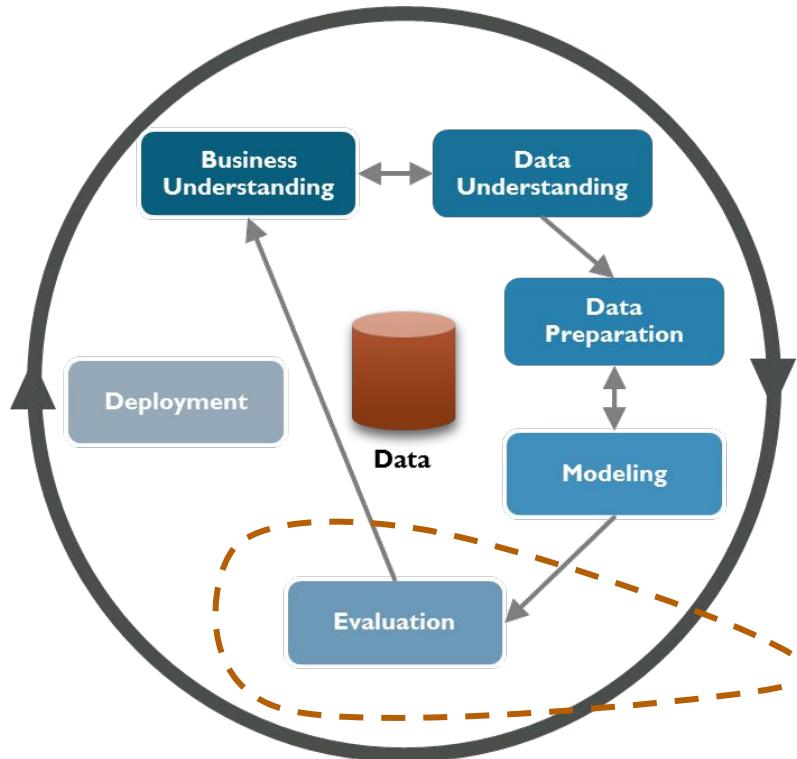
No âmbito da ciência de dados, o padrão também é chamado de **hipótese**



Esta outra hipótese também modela perfeitamente os **dados de treinamento**

Será que ela é melhor?

Como funciona um projeto de Aprendizado de máquina



CRISP - DM

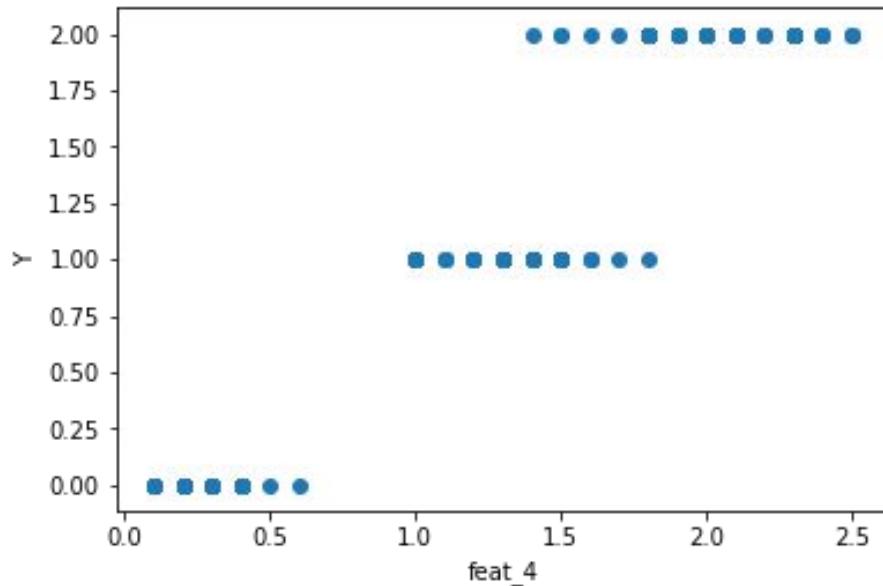
O exemplo anterior
mostra a
necessidade de
métricas de
avaliação dos
modelos

Visualização de dados

Como sugerido pelo exemplo da especificação de imóveis, uma forma do cientista de dados ter ideias a respeito de quais **hipóteses** testar é a **visualização dos dados**

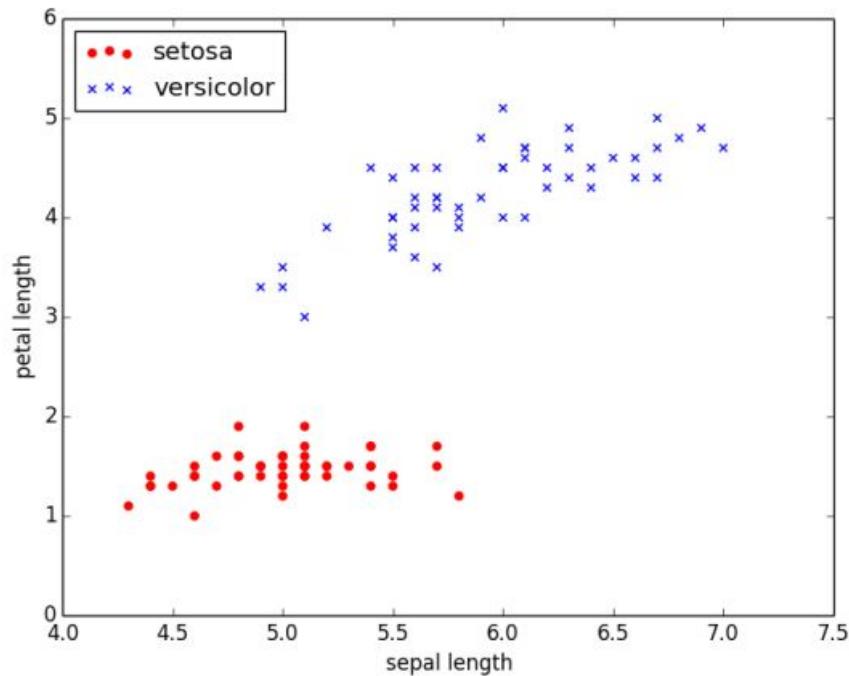
Visualização de dados

Uma possibilidade é dispor a relação entre **features individuais** e a **variável objetivo** em um gráfico de dispersão ou **scatter plot**



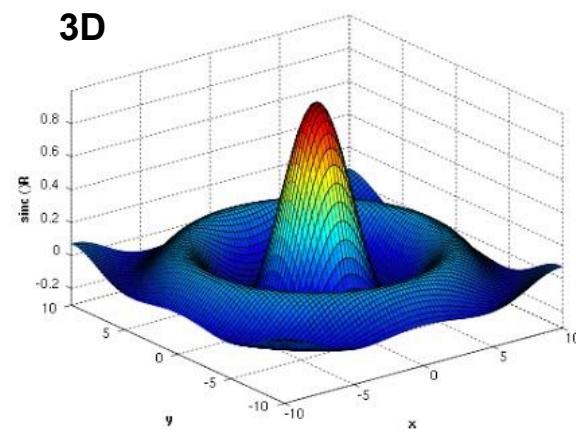
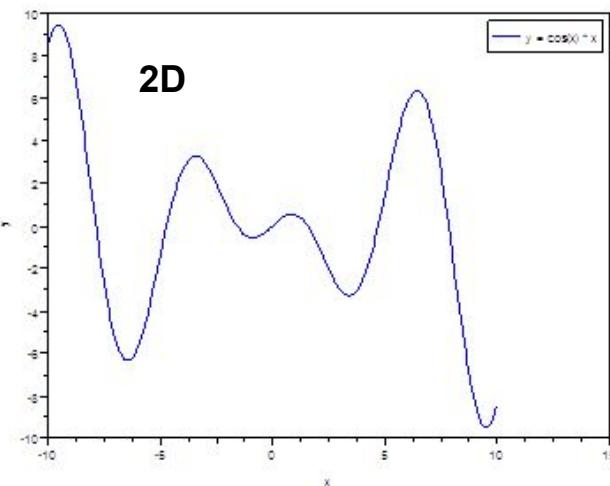
Visualização de dados

Outra possibilidade é analisar a relação entre duas variáveis e representar a variável objetivo pelo tipo de marcação



Visualização de dados - TSNE

Seria possível analisar através de um gráfico 2d pontos que estão em um hiperplano de dimensão maior que 2?

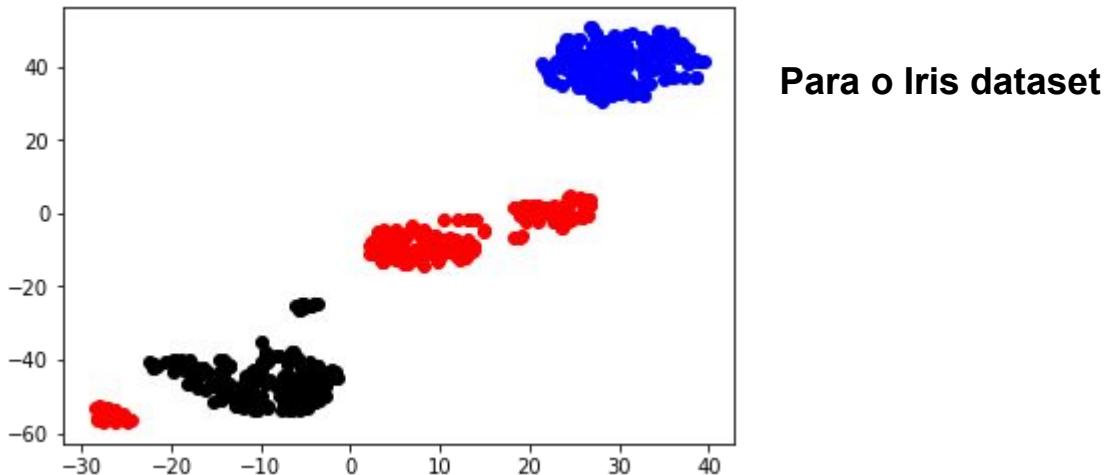


20D

?

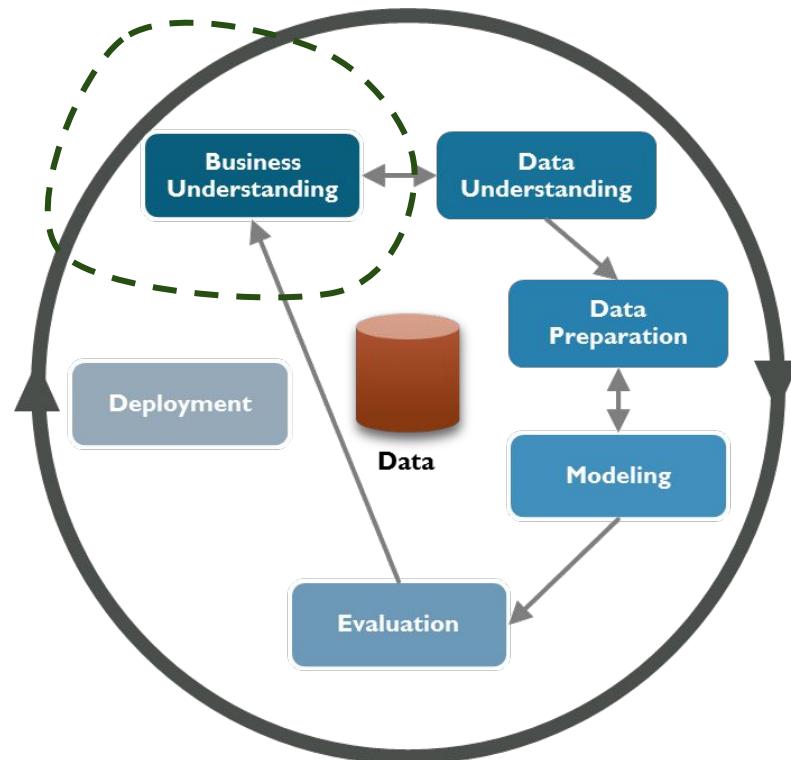
Visualização de dados - TSNE

A técnica **TSNE** se propõe a representar um reticulado hiperdimensional por pontos 2d preservando as relações de proximidade. Basicamente com o TSNE podemos ter uma idéia se as features utilizadas são ou não satisfatórias para a classificação



Como funciona um projeto de Aprendizado de máquina

Neste momento,
avaliamos se o
desempenho do
modelo é adequado
ao negócio



CRISP - DM

Por que utilizar Python?

Python é uma linguagem de alto nível interpretada e de propósito geral.

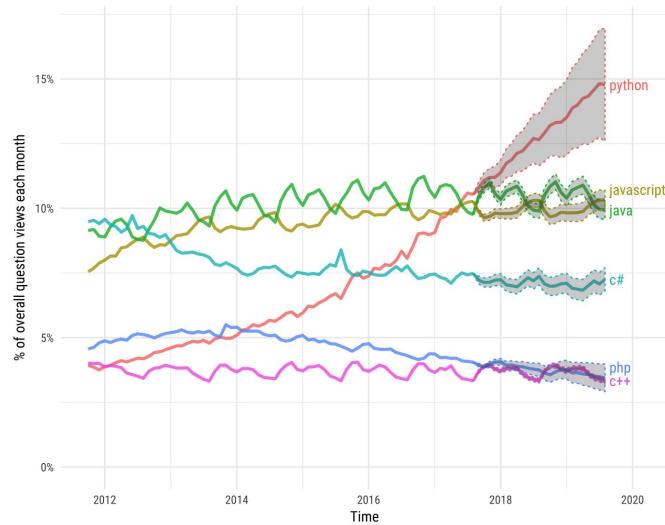
É também **multi-plataforma**, **multi-paradigma**, além de possuir tipagem dinâmica e gerenciamento automático de memória.

É uma das linguagens **mais utilizadas** para aplicações de aprendizado de máquina



Projections of future traffic for major programming languages

Future traffic is predicted with an STL model, along with an 80% prediction interval.



Por que utilizar Python?

Python é uma linguagem de **fácil leitura** e possui muitos pacotes para lidar com **diversos tipos de dados** de forma simples

Imagen



Áudio



Texto



Dados tabulares



Por que utilizar Python?

O mesmo é verdade para pacotes de análise de dados

Análise numérica



Visualização de dados



Mineração e leitura de dados



Aprendizado de máquina



Deep Learning



Testando Python no Browser

Ao longo do curso, utilizaremos uma ferramenta simples para testar o Python e suas principais bibliotecas em um ambiente visual



Gerenciando dependências com o Python

O Python possui um gerenciador de dependências bastante simples chamado **pip** os principais comandos são:

pip list

pip install <package>

pip install <package>==<version>

pip uninstall <package>

pip install -r requirements.txt

É possível criar um documento com as dependências da seguinte forma:

```
numpy == 1.14.5
pandas == 0.23.1
scikit-learn == 0.19.1
scipy == 1.1.0
python-dateutil == 2.7.3
tqdm == 4.23.4
pydotplus == 2.0.2
sphinx == 1.7.6
matplotlib == 2.2.2
vertica-python == 0.7.3
s3io == 0.1.1
boto3 == 1.9.11
awscli == 1.16.21
torch == 0.4.0
torchvision == 0.2.1
```

Análise numérica em Python

Os conhecimentos matemáticos mais importantes para a análise de dados são a álgebra linear e a estatística. Em função disso, surgiu a biblioteca **Numpy**

Exemplo de entendimento dos dados

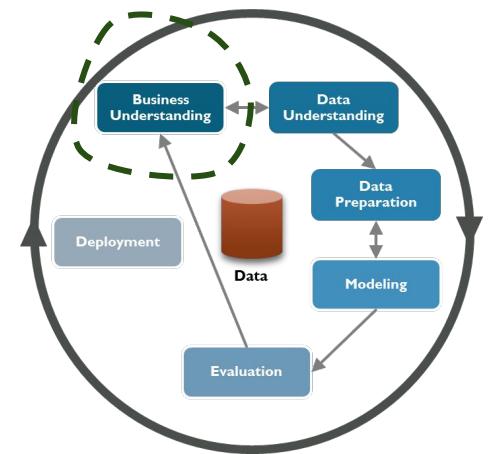
Análise da qualidade das instituições de ensino superior utilizando micrdados do ENADE e do Censo da Educação Superior

Caracterização do problema

Ao longo dos últimos 20 anos, o número de instituições de educação superior no Brasil mais que dobrou. No entanto:

1. O Brasil ainda é o 60º colocado em educação em um ranking de 76 países criado pela OCDE (Organização para a Cooperação e Desenvolvimento Econômico);

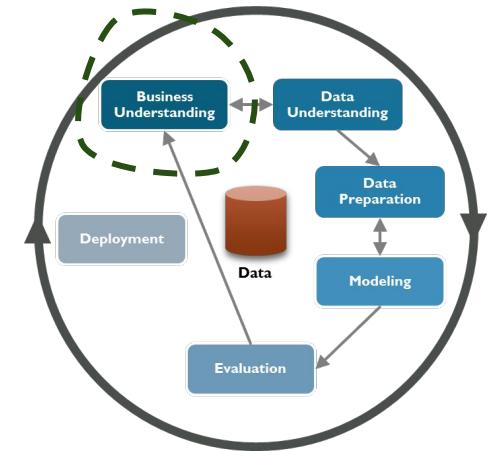
2. O Brasil não possui nenhuma universidade entre as 100 melhores do mundo.



Objetivo

Definir os **fatores que mais influenciam** para a qualidade de um **curso de graduação**.

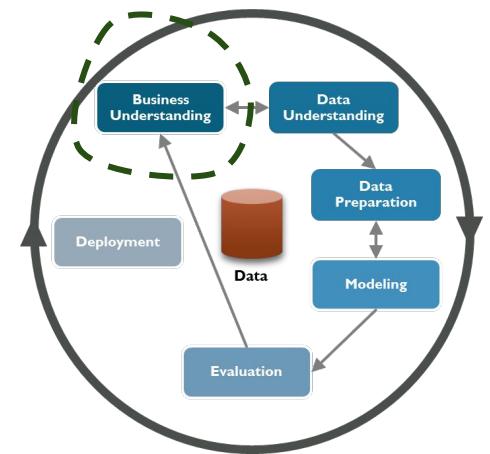
Esses fatores podem influenciar na criação de **diretrizes mais eficientes** para tratar o déficit da educação de nível superior brasileira.



Entendimento do negócio

Definir os **fatores que mais influenciam** para a qualidade de um **curso de graduação**.

Esses fatores podem influenciar na criação de **diretrizes mais eficientes** para tratar o déficit da educação de nível superior brasileira.



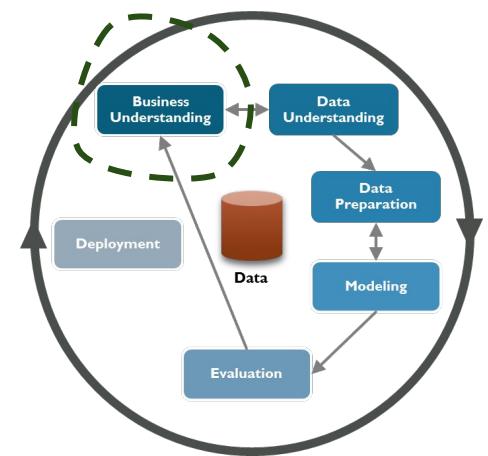
Entendimento do negócio

Como as instituições são avaliadas hoje em dia

Atualmente, o INEP (Instituto Nacional de Estudos e Pesquisas Educacionais) utiliza as seguintes métricas para avaliar os cursos de graduação no Brasil.

1. Notas dos alunos no **ENADE**;
2. Características do corpo docente;
3. Instalações físicas;
4. Organização didático-pedagógica;

Essas informações são integradas no chamado índice **CPC** (Conceito preliminar de curso)

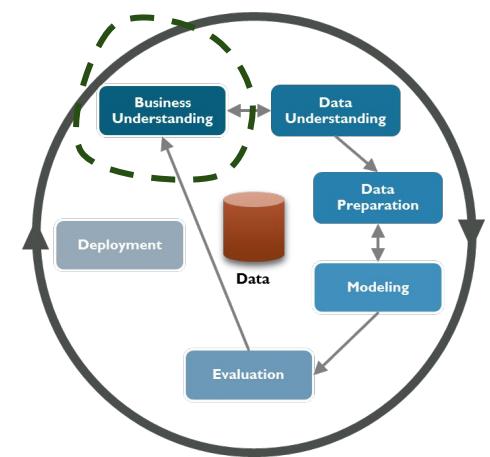


Entendimento do negócio

Definição da métrica de desempenho

Queremos verificar também a influência das características do corpo docente e da organização didático-pedagógica na qualidade dos cursos. Dessa forma, a utilização do CPC como métrica de desempenho carrega **informação à posteriori**.

Assim, podemos utilizar apenas a nota do **ENADE**, uma vez que ele já é **utilizado** para avaliar as escolas implicitamente no CPC.



Entendimento do negócio

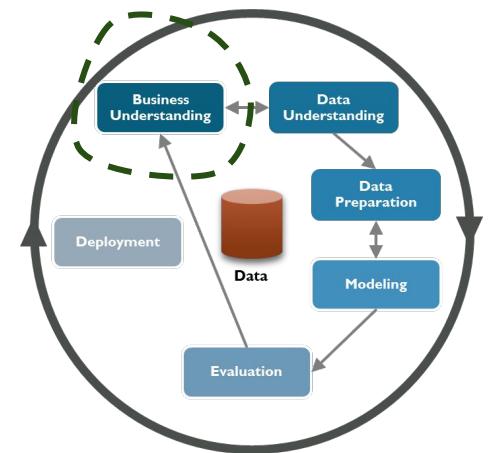
ENADE

Prova destinada à avaliação de instituições de ensino superior, obrigatória para os alunos selecionados e **condição indispensável** para a emissão de um histórico escolar.

O ENADE possui um **Ciclo de Avaliação**, onde a cada ano, apenas algumas áreas do conhecimento se submetem à prova.

Últimos dados disponíveis: ENADE 2014

Avaliaremos apenas as áreas do conhecimento que **prestaram a prova em 2014**.



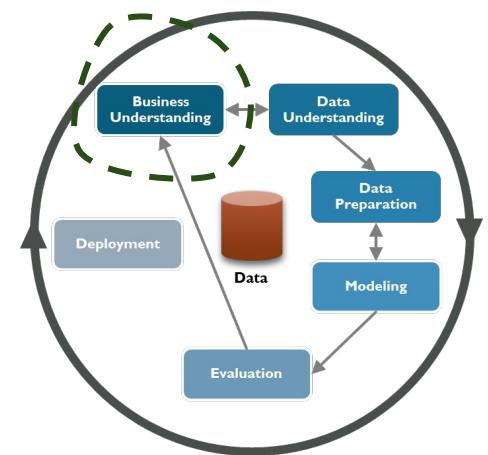
Entendimento do negócio

Definição do grão de análise

Assim como o INEP, queremos avaliar as instituições respeitando as **diferenças entre os cursos**.

GRÃO DE ANÁLISE (VERSAO 1): O par Instituição x curso, exemplo:

UFPE - Engenharia Eletrônica; UFBA - Ciência da Computação



Entendimento dos dados

Base de dados disponível

Censo
Escolar

Cerca de 5GB de dados

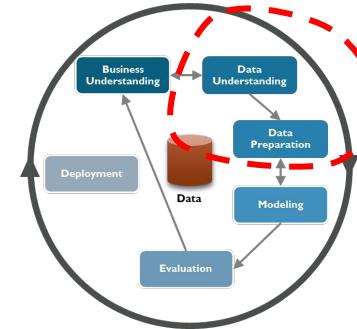
Dados de várias granularidades

- Instituição: 2.368
- Curso: 33.274
- Aluno 10.793.933
- Professor: 396.596

ENADE

481.721 registros

Dados na granularidade aluno



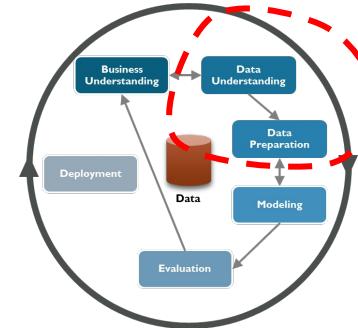
- Existe um índice comum entre as duas bases de dados para identificar a instituição de ensino (IES);
- Não existe um índice comum para a identificação do curso;
- Nas duas bases de dados, existem campos que identificam a **área do curso**;

GRÃO DE ANÁLISE (Versão 2): Instituição x área do conhecimento:

Ex : UFPE - Humanas / UFPE - Engenharia / USP - Educação, etc....

Entendimento dos dados

Área dos cursos - Base do Censo Escolar



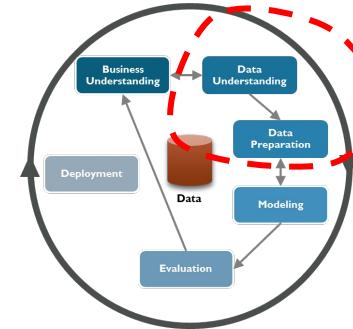
São utilizados códigos OCDE (Organização para Cooperação e Desenvolvimento Econômico). O primeiro dígito define a área geral do conhecimento:

1	Educação
2	Humanidades e Artes
3	Ciências Sociais, Negócios e Direito
4	Ciências, Matemáticas e Computação
5	Engenharia, Produção e Construção
6	Agricultura e Veterinária
7	Saúde e bem estar social
8	Serviços

Entendimento dos dados

Área do curso - Base do ENADE

21 = ARQUITETURA E URBANISMO
 72 = TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
 73 = TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL
 76 = TECNOLOGIA EM GESTÃO DA PRODUÇÃO INDUSTRIAL
 79 = TECNOLOGIA EM REDES DE COMPUTADORES
 701 = MATEMÁTICA (BACHARELADO)
 702 = MATEMÁTICA (LICENCIATURA)
 903 = LETRAS-PORTUGUÊS (BACHARELADO)
 904 = LETRAS-PORTUGUÊS (LICENCIATURA)
 905 = LETRAS-PORTUGUÊS E INGLÊS (LICENCIATURA)
 906 = LETRAS-PORTUGUÊS E ESPANHOL (LICENCIATURA)
 1401 = FÍSICA (BACHARELADO)
 1402 = FÍSICA (LICENCIATURA)
 1501 = QUÍMICA (BACHARELADO)
 1502 = QUÍMICA (LICENCIATURA)
 1601 = CIÊNCIAS BIOLÓGICAS (BACHARELADO)
 1602 = CIÊNCIAS BIOLÓGICAS (LICENCIATURA)
 2001 = PEDAGOGIA (LICENCIATURA)
 2401 = HISTÓRIA (BACHARELADO)
 2402 = HISTÓRIA (LICENCIATURA)
 2501 = ARTES VISUAIS (LICENCIATURA)
 3001 = GEOGRAFIA (BACHARELADO)
 3002 = GEOGRAFIA (LICENCIATURA)
 3201 = FILOSOFIA (BACHARELADO)
 3202 = FILOSOFIA (LICENCIATURA)
 3502 = EDUCAÇÃO FÍSICA (LICENCIATURA)
 4004 = CIÉNCIA DA COMPUTAÇÃO (BACHARELADO)
 4005 = CIÉNCIA DA COMPUTAÇÃO (LICENCIATURA)
 4006 = SISTEMAS DE INFORMAÇÃO
 4301 = MÚSICA (LICENCIATURA)
 5401 = CIÉNCIAS SOCIAIS (BACHARELADO)
 5402 = CIÉNCIAS SOCIAIS (LICENCIATURA)
 5710 = ENGENHARIA CIVIL
 5806 = ENGENHARIA ELÉTRICA
 5809 = ENGENHARIA DE COMPUTAÇÃO
 5814 = ENGENHARIA DE CONTROLE E AUTOMAÇÃO
 5902 = ENGENHARIA MECÂNICA
 6008 = ENGENHARIA QUÍMICA
 6009 = ENGENHARIA DE ALIMENTOS
 6208 = ENGENHARIA DE PRODUÇÃO
 6306 = ENGENHARIA
 6307 = ENGENHARIA AMBIENTAL
 6405 = ENGENHARIA FLORESTAL



Realizamos a conversão dos códigos utilizados no ENADE para os códigos OCDE através de um dicionário.

Entendimento dos dados

Base do Censo Escolar

Esta base possui dados em vários grãos

DM_ALUNO: Grão aluno,

Identificação dos cursos/ área/ sexo / raça / idade / necessidades especiais / participação em pesquisas ou projetos assistencialistas / etc...;

DM_CURSO: Grão curso,

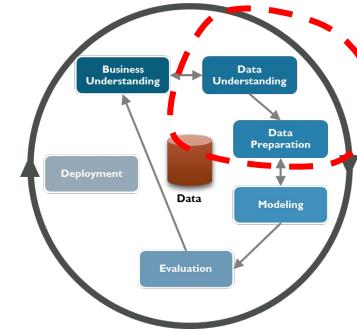
Identificação dos cursos/ área/ carga horária/ recursos didáticos e acessibilidade/ vagas e métodos de ingresso/ número de concluintes e ingressantes;

DM_DOCENTE: Grão docente (está separado por IES, não por curso)

Escolaridade / necessidades especiais;

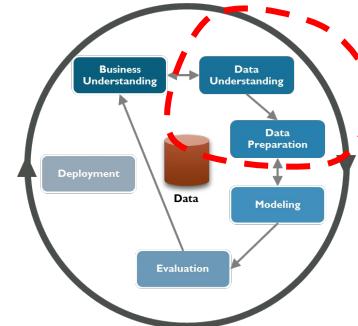
DMIES: Grão instituição de ensino;

Código da Instituição / Quantidade de Técnicos de todos os níveis / Valores de Receitas (Transferências) em R\$



Entendimento dos dados

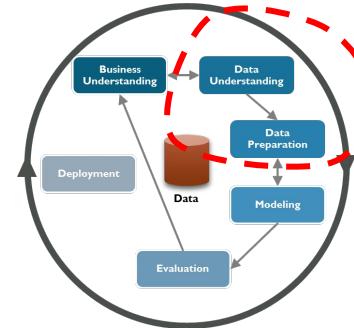
Base ENADE



Grão aluno. Utilizamos apenas a informação da nota média, pois os demais dados já estão presentes em DM_ALUNO com melhor qualidade

Pré processamento dos dados

Missing data

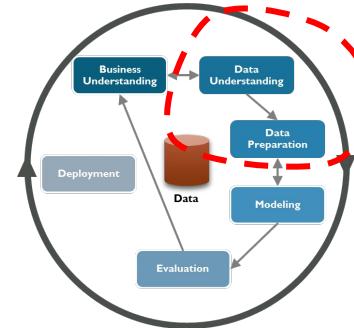


Missing data para valores categóricos: Criação de uma nova categoria para missing;

Missing para valores numéricos: Substituição pela média dos demais (variância artificial)

Pré processamento dos dados

Aglutinação dos dados



Variáveis categóricas: Criação de variáveis dummy e aglutinação pela média, assim, no grão final, temos a proporção de elementos em cada uma das classes;

Valores monetários: Aglutinação pela média;

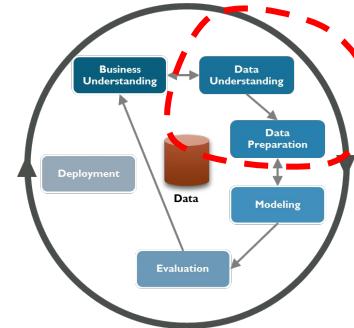
Variáveis numéricas que representam contagens ou valores contínuos (ex: Número de vagas EAD ou idade do aluno): Aglutinação pela média;

Datas (ex: Data de abertura do curso): Consideramos apenas o ano e tomamos a média;

Idade média = Média(ano atual - ano de abertura) = ano atual - Média (ano de abertura)

Pré processamento dos dados

Colunas removidas



A fim de garantir que o classificador construído não aprenda aspectos regionais da educação, removemos as colunas relativas à localização das IES, ex: UF, Estado, etc...

Para alunos estrangeiros, consideramos apenas a booleana: é brasileiro ou não, ignoramos o país de origem por possuir muitas categorias

Pré processamento dos dados

Mudança de grão e filtragem de dados

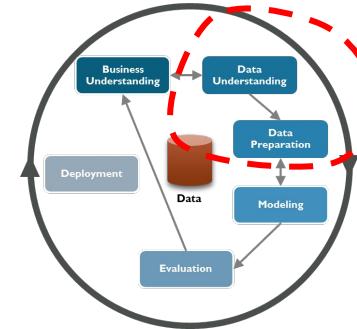
DM_ALUNO

10793935 (total)

6809245 (aluno cursando)

6786979
(co_ocde_area_geral
presente)

perda total: 37.12 %



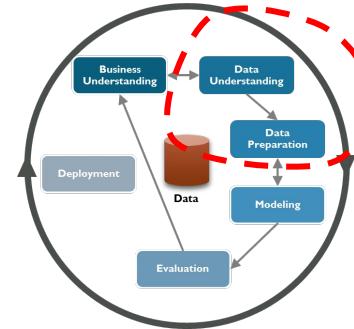
agrupamento
IES / Área

7222
registros

Pré processamento dos dados

Mudança de grão e filtragem de dados

DM_CURSO



33273 (total)

31069 (curso ativo e com ano de inicio)

30868(co_ocde_area_geral presente)

perda total: 7.22 %

agrupamento
IES / Área

7215
registros

Pré processamento dos dados

Mudança de grão e filtragem de dados

DM_DOCENTE

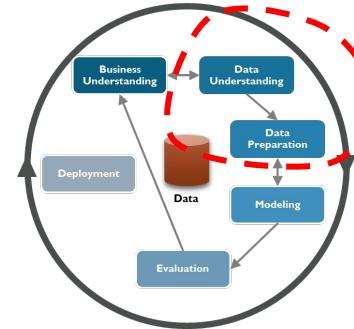
396595 (total)

383386 (docente em atividade)

perda total: 3.33 %

agrupamento
IES

2368
registros



Pré processamento dos dados

Mudança de grão e filtragem de dados

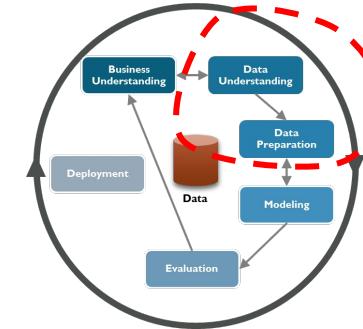
DM_ENADE

481720 (total)

395557 (presente)

395453 (nota geral
presente)

perda total: 17.90 %



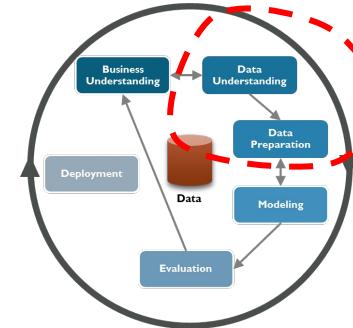
agrupamento
IES / Área

2626
registros

Pré processamento dos dados

Junção de dados

IES presentes no Enade mas não no Censo....



ALUNOS
7222

Enade
2626

Curso
7215

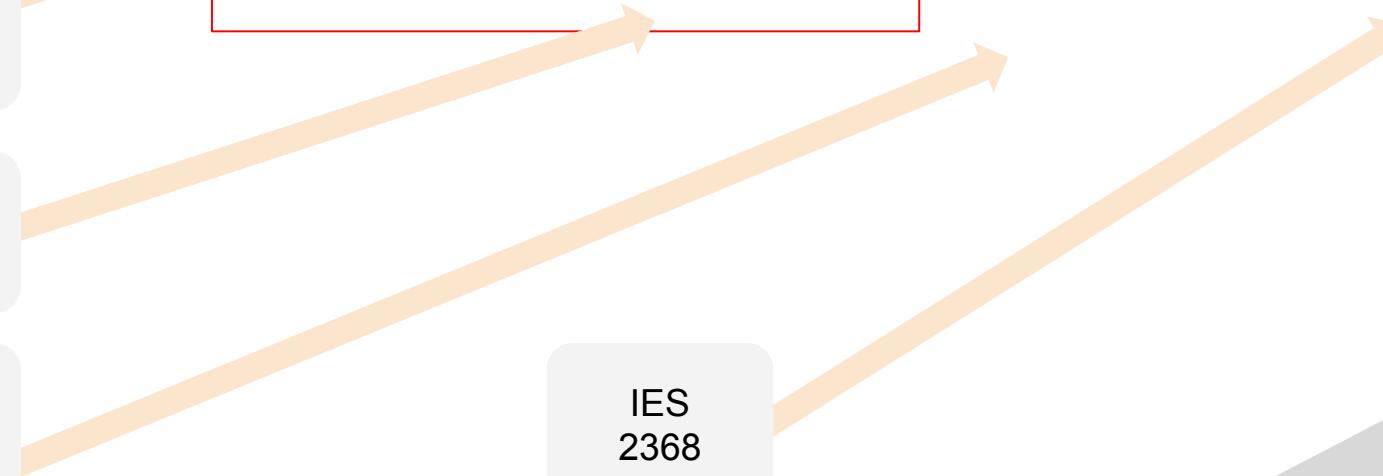
Docente
2368

Join 1
2593

Join 2
2580

Join 3
2580

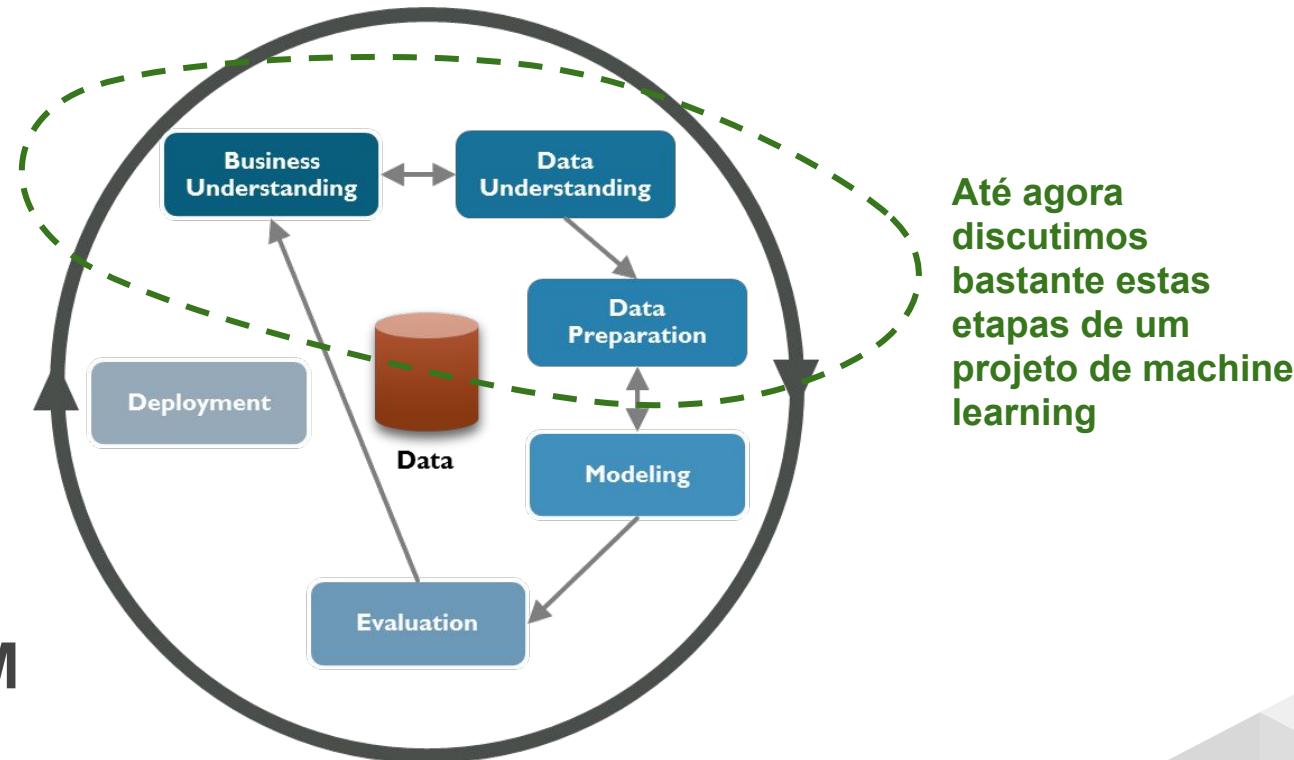
Join 4
2580



Avalie a primeira parte do curso

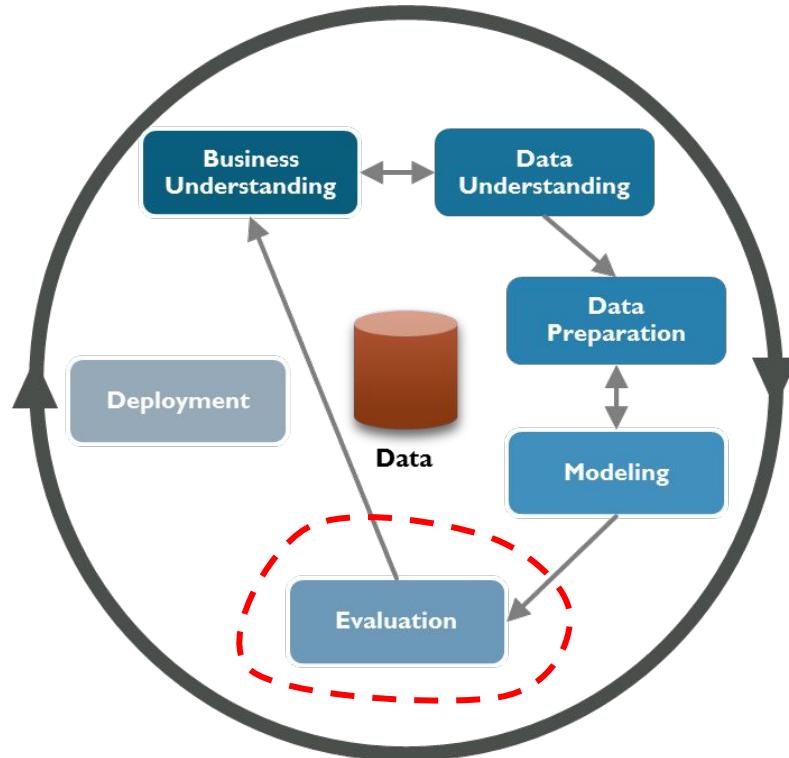
https://docs.google.com/forms/d/e/1FAIpQLScjRfErajmoXcInExnMia32RJ9NLDQbtS_DJ25jGHYDSmhQbg/viewform?usp=sf_link

Como funciona um projeto de Aprendizado de máquina



CRISP - DM

Como funciona um projeto de Aprendizado de máquina

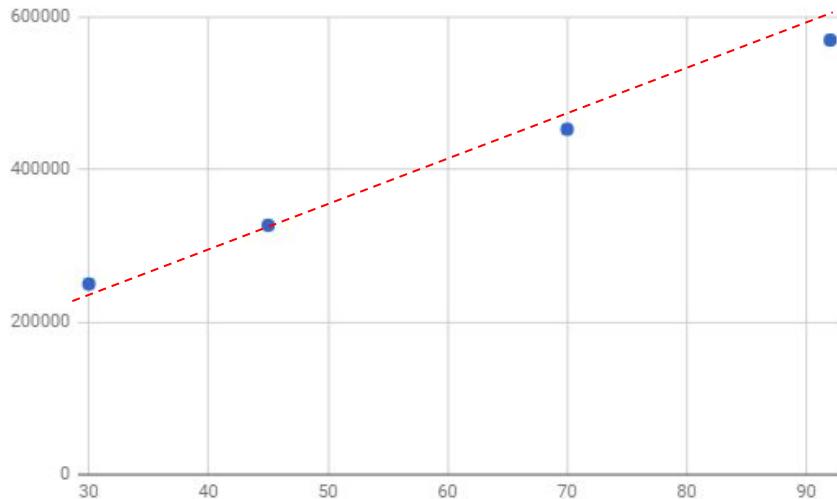


CRISP - DM

Pularemos por enquanto a modelagem e vamos falar de como sabemos se um modelo é bom ou não

Métricas de desempenho

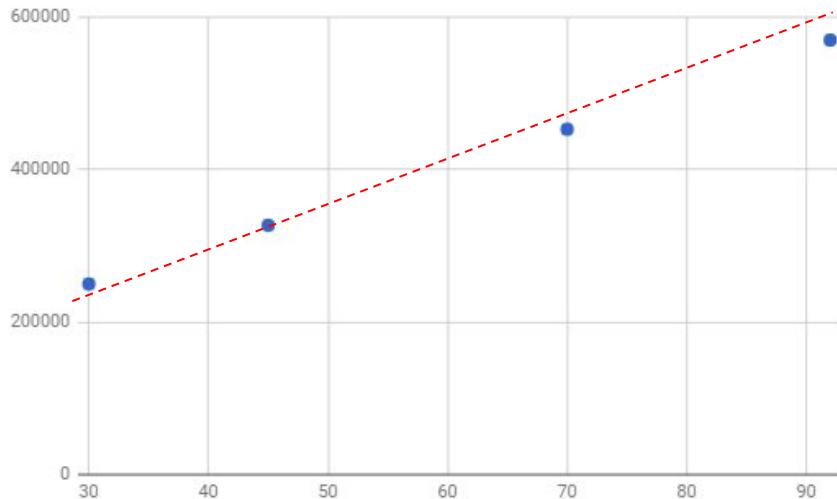
Voltando ao exemplo da precificação de imóveis, como selecionamos o melhor modelo?



Neste caso, a nossa hipótese é que a lei que regia o fenômeno era uma linha reta, mas **precisava ser?**

Métricas de desempenho

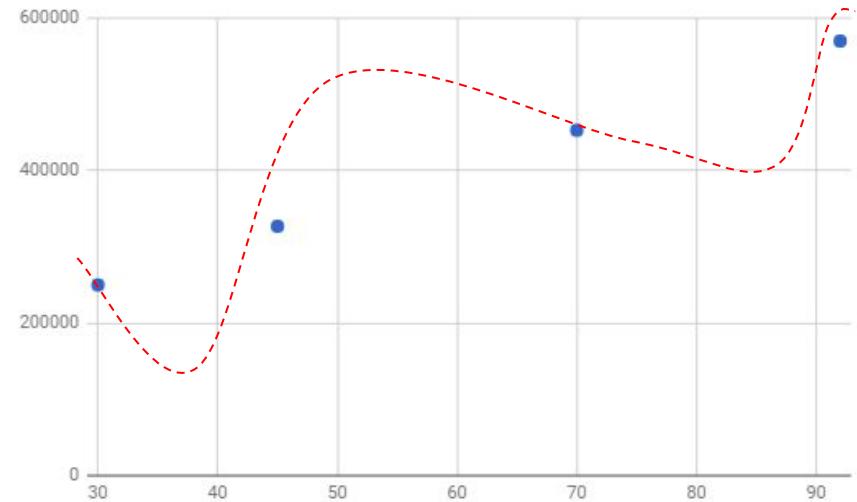
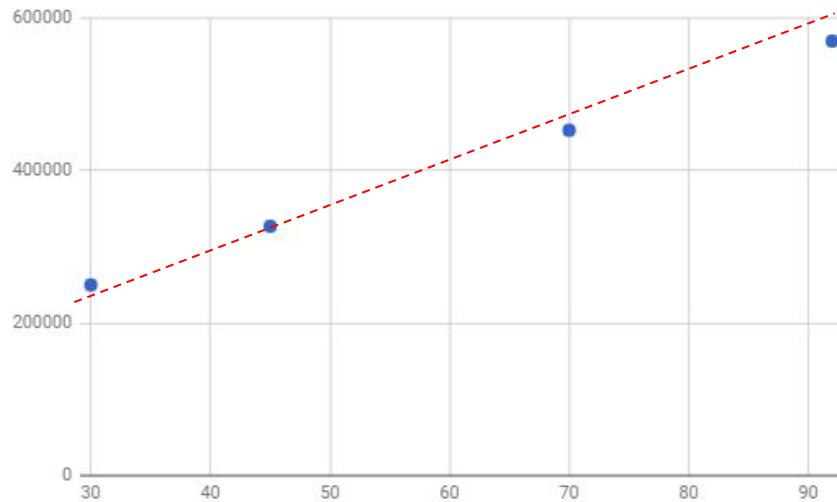
Voltando ao exemplo da precificação de imóveis, como selecionamos o melhor modelo?



Neste caso, a nossa hipótese é que a lei que regia o fenômeno era uma linha reta, mas **precisava ser?**

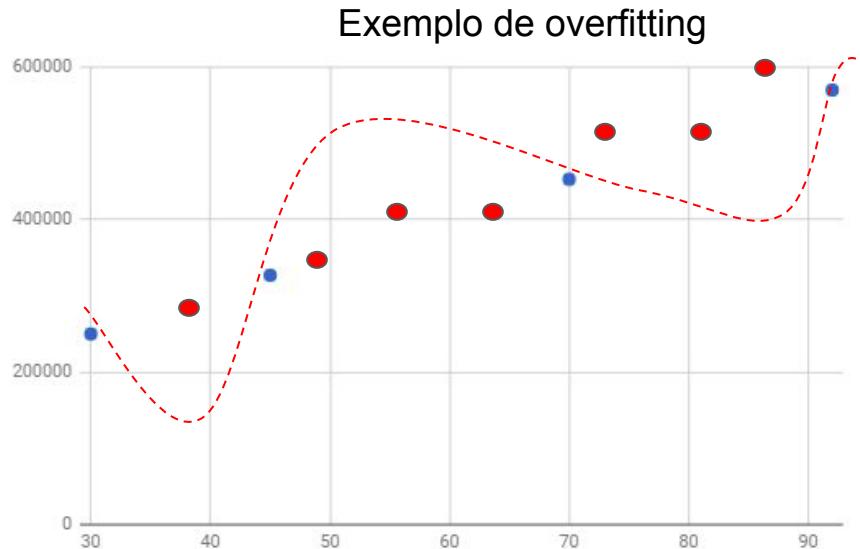
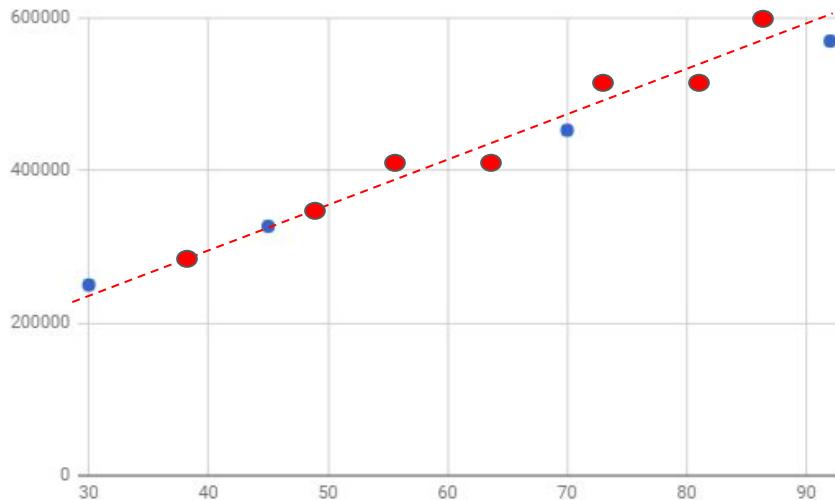
Métricas de desempenho

No conjunto de **treinamento** as duas hipóteses parecem igualmente interessantes



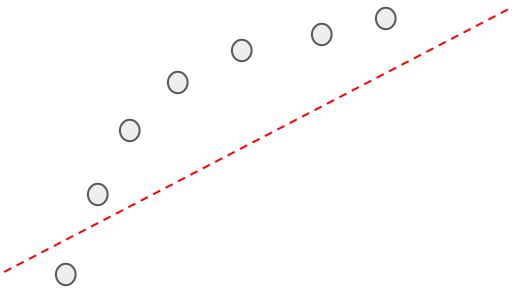
Métricas de desempenho

Caso seja utilizado para fins de teste um segundo conjunto de dados que não foi usado no treinamento, podemos perceber que o primeiro modelo é melhor



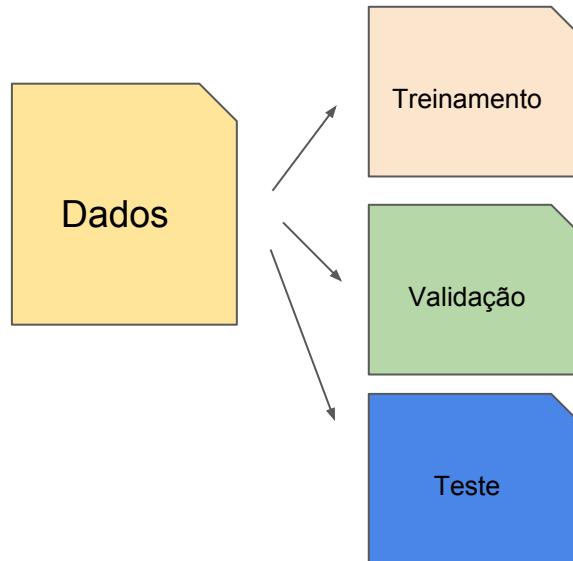
Métricas de desempenho

Underfitting seria o caso contrário



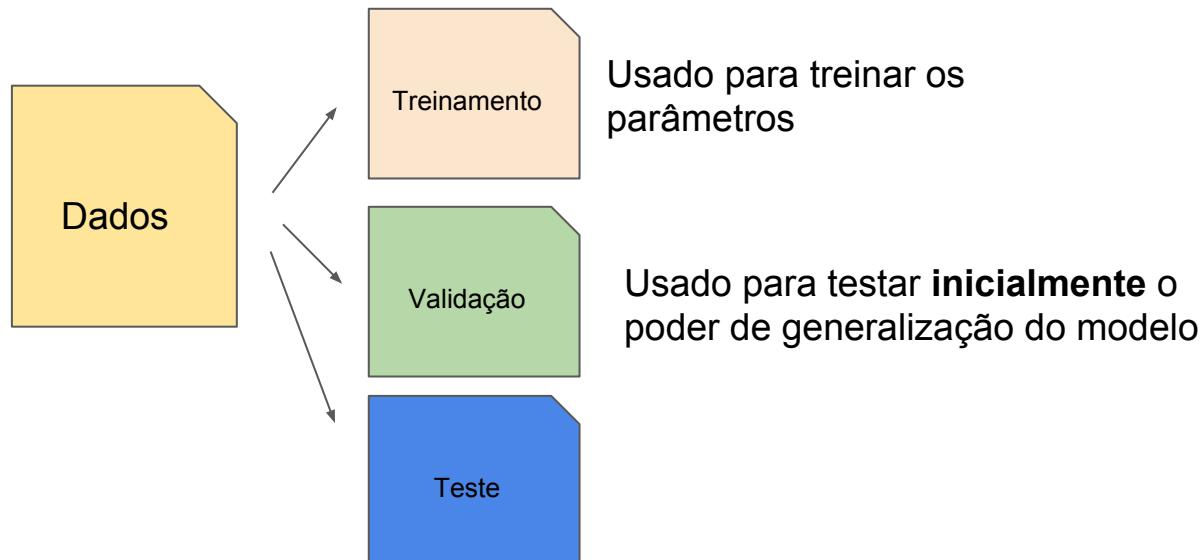
Métricas de desempenho

Em geral, dividimos o banco de dados disponível em três conjuntos, o de **treinamento**, o de **validação** e o de **teste**



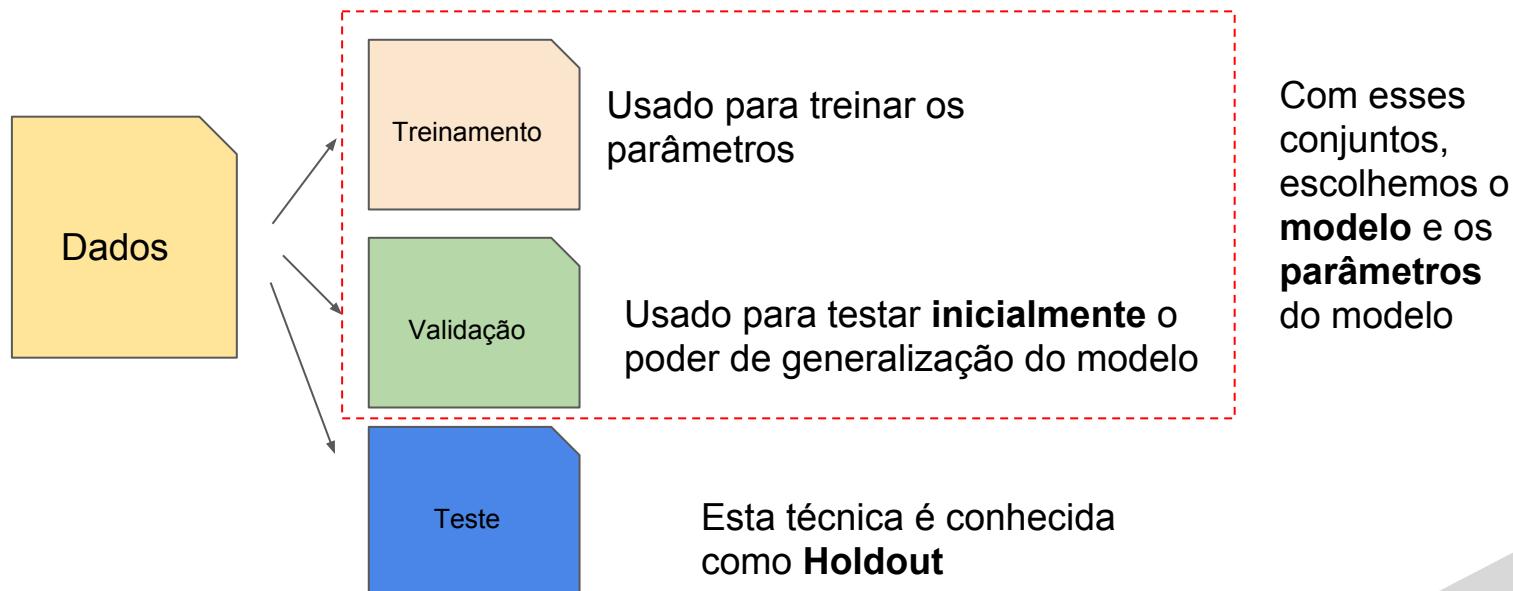
Métricas de desempenho

Em geral, dividimos o banco de dados disponível em três conjuntos, o de **treinamento**, o de **validação** e o de **teste**



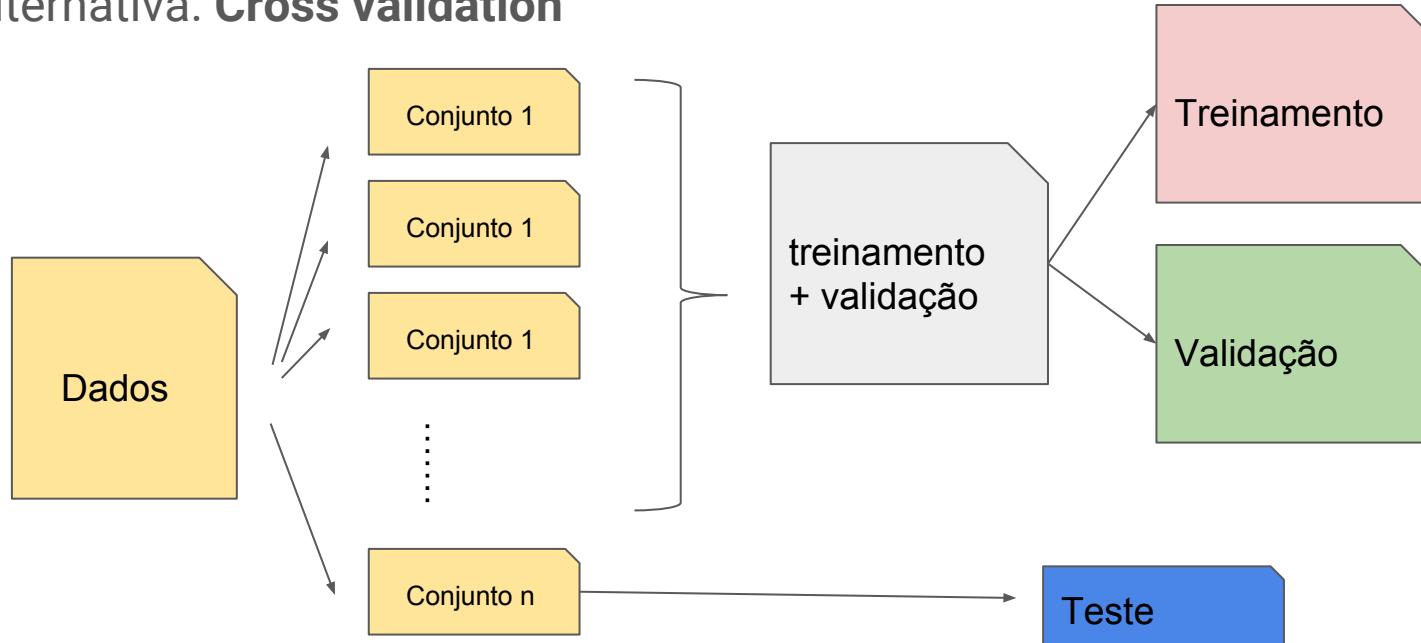
Métricas de desempenho

Em geral, dividimos o banco de dados disponível em três conjuntos, o de **treinamento**, o de **validação** e o de **teste**



Métricas de desempenho

Alternativa: **Cross validation**



Métricas de desempenho

Alternativa: **Cross validation**

Os treinamentos são realizados n vezes variando qual **conjunto será o conjunto de teste**.

O cross validation possui menos variância e é uma boa opção para conjuntos menores pois utiliza todos os dados para treinamento em algum momento.

Métricas de desempenho

São formas de verificar se um determinado algoritmo de machine learning está classificando bem ou não. Não confundir com **função de custo**

Primeira métrica: Acurácia

De todos os testes realizados, quantos resultaram em uma classificação correta?

Acurácia = (número de testes corretos) / (total de testes)

Métricas de desempenho

Problemas da acurácia:

Vamos supor que estamos fazendo um **teste para detectar câncer**

Suponha que 1 em cada 100 pessoas no mundo desenvolvem câncer,
um classificador que sempre diz que a pessoa não tem câncer terá

99% de acurácia!

Será que este classificador é bom?

Métricas de desempenho

Uma forma de conhecer melhor o desempenho do sistema é a matriz de confusão. Suponha que temos um problema de classificação binária (classe A e classe B). Podemos montar a seguinte matriz

	A	B
A	Verdadeiros positivos da classe A	Falsos negativos da classe A / Falsos positivos da classe B
B	Falsos negativos da classe B / Falsos positivos da classe A	Verdadeiros positivos da classe B

Métricas de desempenho

De posse da matriz de confusão, podemos definir algumas outras métricas

	A	B
A	TP_a	FP_b
B	FP_a	TP_b

Precisão da classe A: $TP_a / (TP_a + FP_a)$

Precisão da classe B: $TP_b / (TP_b + FP_b)$

Métricas de desempenho

De posse da matriz de confusão, podemos definir algumas outras métricas

	A	B
A	TP_a	FP_b
B	FP_a	TP_b

Recall da classe A: $TP_a / (TP_a + FP_b)$

Recall da classe B: $TP_b / (TP_b + FP_a)$

Note que o recall é igual à **acurácia por classe**

Métricas de desempenho

De posse da matriz de confusão, podemos definir algumas outras métricas

	A	B
A	TP_a	FP_b
B	FP_a	TP_b

Podemos também definir precisão e recall médios

Precisão média: $\{(número\ de\ A) * P_a + (número\ de\ B) * P_b\} / \{(número\ de\ A) + (número\ de\ B)\}$

Recall médio: $\{(número\ de\ A) * R_a + (número\ de\ B) * R_b\} / \{(número\ de\ A) + (número\ de\ B)\}$

Métricas de desempenho

A precisão e o recall por classe são mais indicadas para avaliar classificadores desbalanceados

	A (câncer)	B (não câncer)
A (câncer)	0	10
B (não câncer)	0	1000

Acurácia: $1000/1010 = 99\%$

Precisão da classe A: 0%

Recall da classe A: 0%

Precisão da classe B: $1000/1010 = 99\%$

Recall da classe B: 100%

Precisão média: $\{99*1000 + 0*10\} / 1010 = 90\%$

Recall média: 99%

Métricas de desempenho

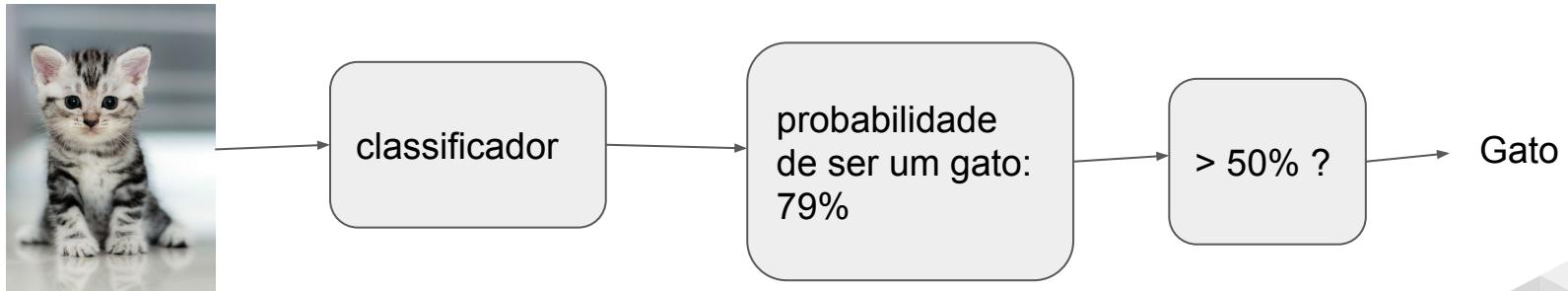
Muitas vezes é utilizada a métrica F1-score como medida

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Métricas de desempenho (resposta contínua)

Muitos classificadores, além de permitirem saber qual a classe de um novo exemplo, também retornam uma **métrica de probabilidade** daquele exemplo de fato ser da classe escolhida.

Por default, consideramos um exemplo como pertencente de uma determinada classe se esta probabilidade é acima de 50%

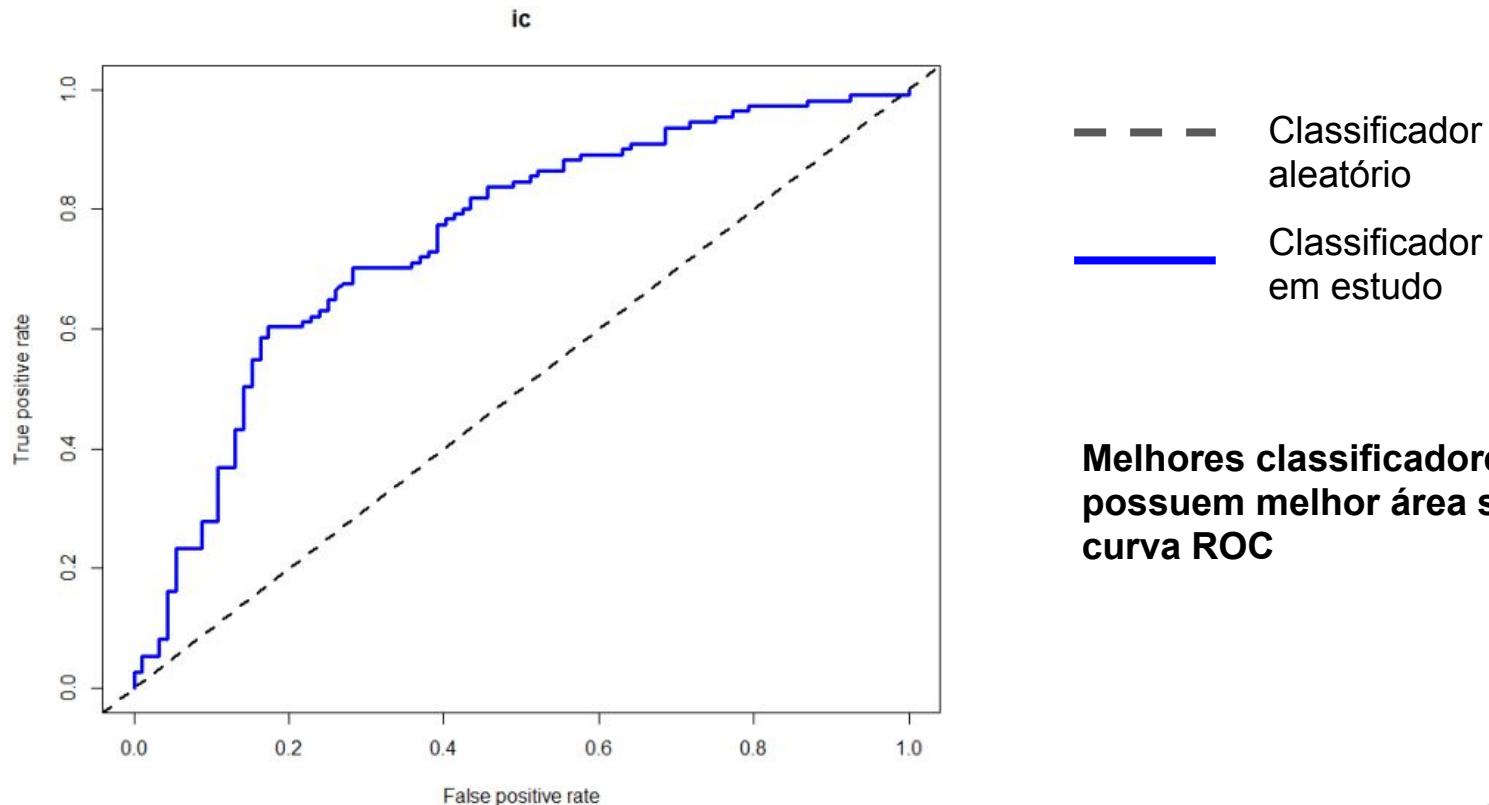


Métricas de desempenho (resposta contínua)

No entanto, podemos decidir ativar a decisão com um limiar **acima de 50%**, quando queremos apenas detecções de alta qualidade, ou **abaixo de 50%** quando desejamos mitigar falsos negativos, por exemplo, em uma detecção de fraude em cartões de crédito.

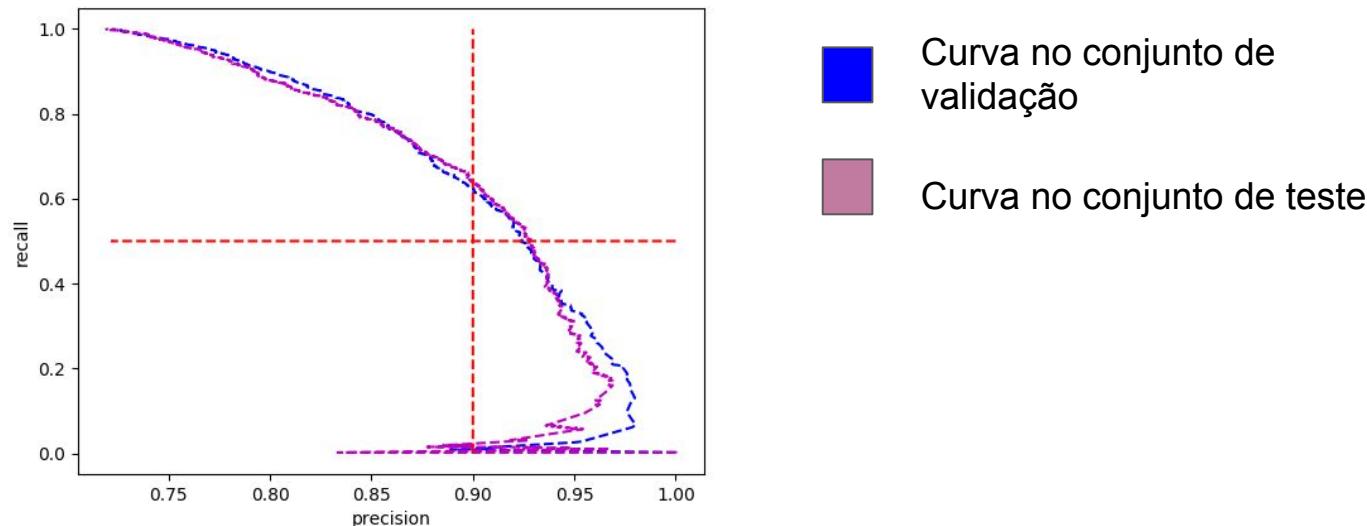
Para cada um desses limiares, podemos calcular a taxa de **verdadeiros positivos e falsos positivos**. Tais pontos podem ser dispostos em uma curva chamada curva ROC. Como comparar classificadores ao longo de **todos os limiares**?

Métricas de desempenho (resposta contínua)



Métricas de desempenho (resposta contínua)

Outra coisa que pode ser feita é construir uma curva de precisão x recall para cada limiar



Noções de intervalo de confiança

Um dos maiores problemas do aprendizado de máquina é saber o quanto **significativas** são as métricas calculadas.

Como exemplo, vamos supor que você deseje saber qual a altura média dos brasileiros (mais de 200 milhões de pessoas). O procedimento padrão é aferir a média de altura em uma **amostra de pessoas**, por exemplo, 1000 pessoas.

Ou seja, estamos querendo tirar conclusões de uma **população** a partir de uma **amostra**

Noções de intervalo de confiança

Precisamos saber o quanto a média dessa amostra se aproxima da verdadeira média da população brasileira (suponha que seja 1.70 m). Se executarmos o experimento em diversas amostras de 1000 pessoas, obteremos diferentes resultados:

amostra 1 : 1.72 m

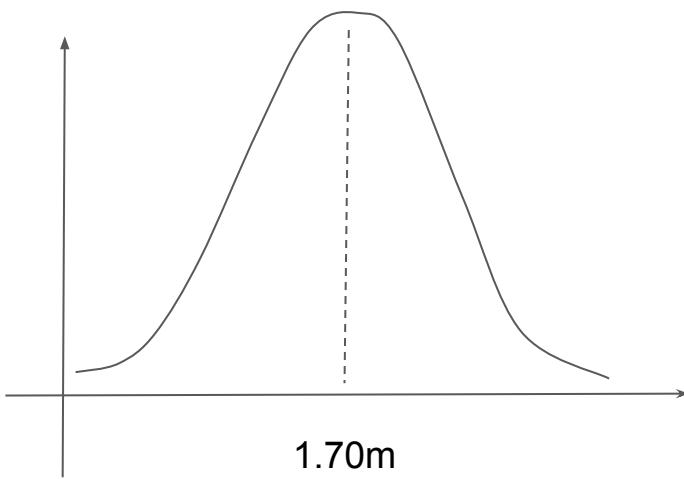
amostra 2: 1.69 m

amostra 3: 1.74 m

.....

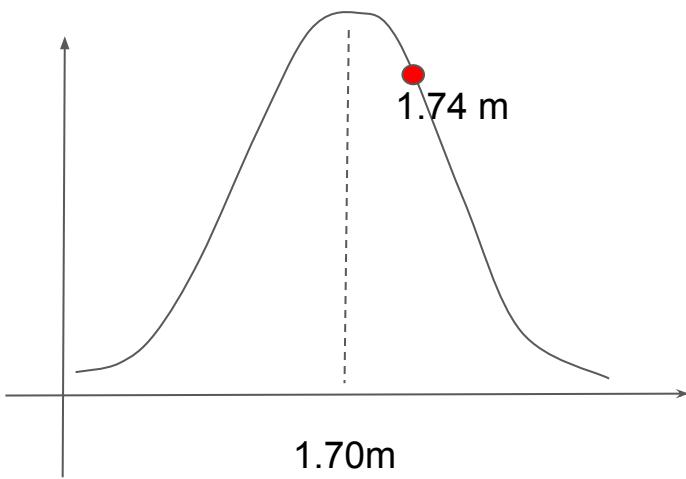
Noções de intervalo de confiança

Caso a distribuição das alturas do brasileiro seja **normal** ou a amostra seja grande o suficiente, a tendência é que as médias observadas tenham igualmente uma distribuição normal em torno da média real



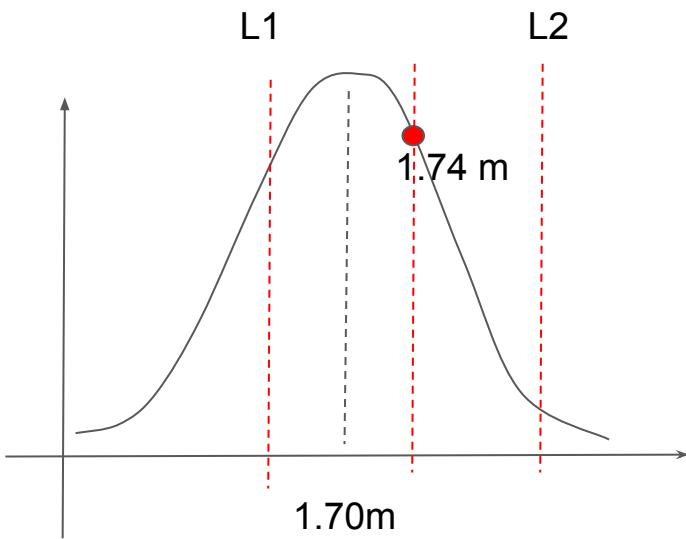
Noções de intervalo de confiança

Com apenas uma amostra, não teremos o valor da média real, mas apenas um ponto na distribuição



Noções de intervalo de confiança

Com apenas uma amostra, não teremos o valor da média real, mas apenas um ponto na distribuição



No entanto, podemos calcular a probabilidade da média real estar distante da média amostral de uma certa margem de erro

$$P(L1 < \text{margem real} < L2)$$

Noções de intervalo de confiança

Da mesma forma que podemos calcular a probabilidade da média real estar dentro de um intervalo, podemos, de forma inversa, calcular **qual o intervalo** onde a média real está com uma probabilidade acima de 90%, 95%, etc...

A este intervalo, damos o nome de **intervalo de confiança**

Noções de intervalo de confiança

A probabilidade do valor real estar no intervalo é chamado de nível de confiança do intervalo, e a sua diferença para 100% é chamada de **nível de significância**, denotada pela letra alfa.

Ex: No intervalo de confiança de 95%, o nível de significância é 5% ou 0.05

Noções de intervalo de confiança

No caso de aprendizado de máquina, queremos calcular intervalos de confiança para proporções (pois acurácia, precisão, recall, etc.. são proporções). Neste caso, para um número n grande de observações, temos que a proporção real p de uma população está no seguinte intervalo com uma probabilidade de $100.(1 - \alpha) \%$

$$\hat{p} - z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \leq p \leq \hat{p} + z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

\hat{p} é a proporção observada na amostra

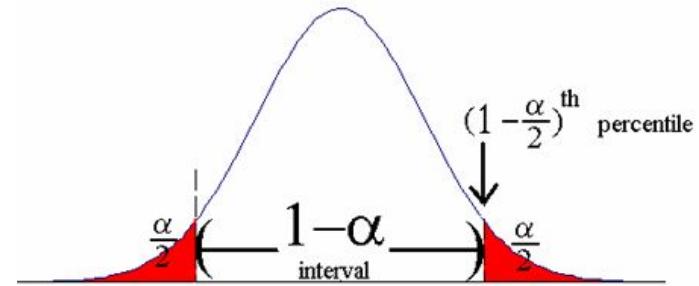
Noções de intervalo de confiança

Os valores de $Z_{\alpha/2}$.

são tabelados, porém, o valor de Zx pode ser obtido facilmente com o python através do seguinte código:

```
import scipy.stats as st
```

```
Zx = st.norm.ppf(1 - x)
```



Noções de intervalo de confiança

Ex: Se em um conjunto de teste de 100 elementos, a acurácia foi de 85%, a acurácia real (do algoritmo em funcionamento no mundo real) deve estar em qual intervalo com 95% de certeza?

$$\alpha \quad 5\% \text{ ou } 0.05$$

$$\hat{p} \quad 0.85$$

$$n \quad 100$$

$$z_{\alpha/2} \quad 1.96$$

$$\hat{p} - z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \leq p \leq \hat{p} + z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

$$0.79 \leq p \leq 0.91$$

Técnicas de classificação

Falaremos aqui de algumas das principais classificadores

- K vizinhos;
- Classificador Bayesiano;
- Regressão logística;
- Multi layer perceptron;
- Support Vector Machines;

K vizinhos

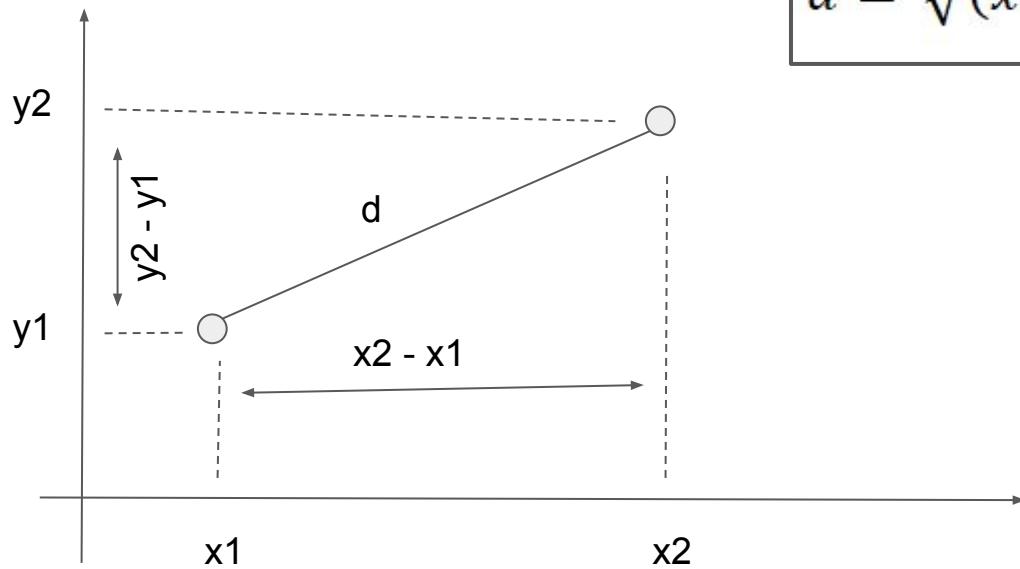
Uma das técnicas de aprendizado de máquina mais simples de ser implementada

Para utilizarmos este algoritmo, basta apenas que todas as variáveis já estejam em forma numérica e de uma **métrica de distância**

Geralmente a métrica de distância utilizada é a **distância euclidiana**

K vizinhos

Distância euclidiana



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

K vizinhos

Distância euclidiana

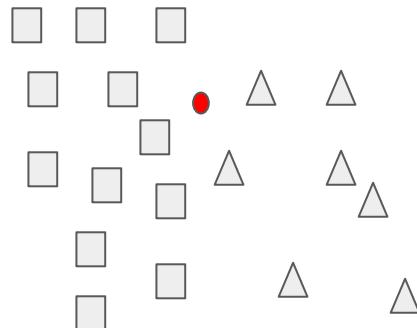
Para **n** variáveis, basta generalizar o conceito:

$$d = \sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2 + (c_2 - c_1)^2 + (d_2 - d_1)^2 + \dots}$$

K vizinhos

Funcionamento

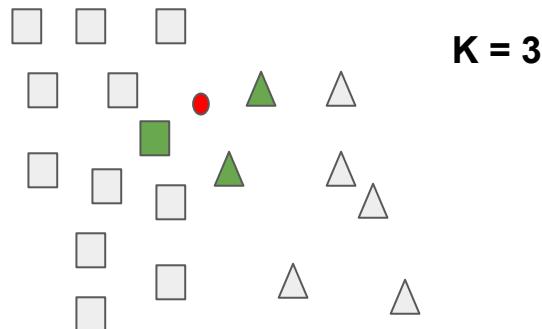
Quando desejamos classificar um ponto novo, olhamos para os K pontos mais próximos (K ímpar) desse ponto na base de dados atual. Damos ao novo ponto a classificação mais **frequente nesses pontos**



K vizinhos

Funcionamento

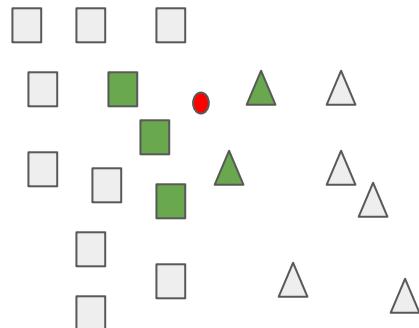
Quando desejamos classificar um ponto novo, olhamos para os K pontos mais próximos (K ímpar) desse ponto na base de dados atual. Damos ao novo ponto a classificação mais **frequente nesses pontos**



K vizinhos

Funcionamento

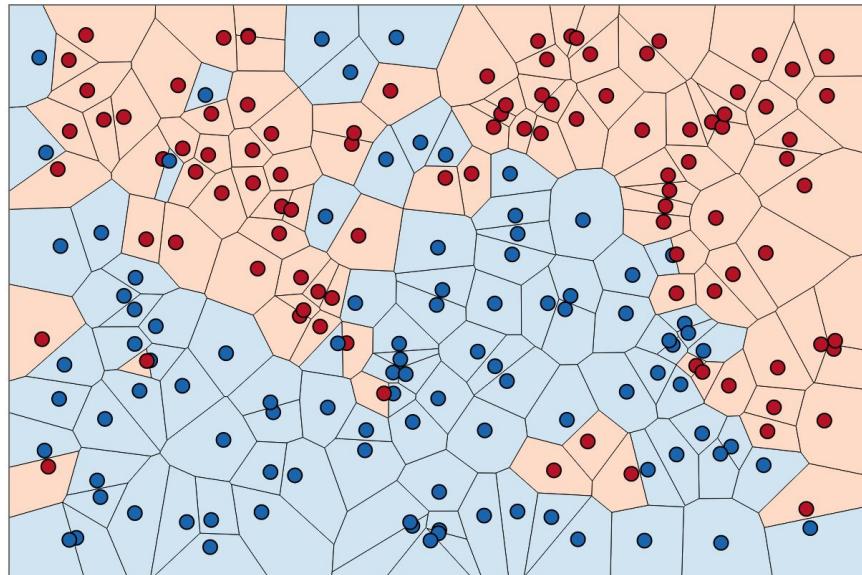
Quando desejamos classificar um ponto novo, olhamos para os K pontos mais próximos (K ímpar) desse ponto na base de dados atual. Damos ao novo ponto a classificação mais **frequente nesses pontos**



É importante validar vários
valores de K diferentes

K vizinhos

Fronteira de decisão



K vizinhos

Vantagens

Fronteira de decisão flexível

Facilidade de implementação

Pode ter aprendizado contínuo

Desvantagens

O aumento da base de dados torna o classificador lento

Overfitting e sensibilidade a dados ruidosos

Classificador Bayesiano

Este classificador procura descrever a densidade de probabilidade dos dados

Exemplo: Temos um classificador que deseja identificar o peixe que está descendo pelo rio

Suponha que existam apenas **dois tipos de peixe**:

Salmão e Bagre

Classificador Bayesiano

Suponha que existe um estudo estatístico que diz que **90%** dos peixes que cruzam o rio são Salmões, logo:

$$P[\text{Salmão}] = 0.9 \text{ e } P[\text{Bagre}] = 0.1$$

Na ausência de qualquer outra informação, o melhor classificador possível é aquele que diz que o peixe é *sempre salmão*

Será que é possível melhorar?

Classificador Bayesiano

Suponha agora que temos um **sensor de luz** que mede o quanto a escama do peixe reflete bem a luz. Será que podemos melhorar o classificador.

Lei da probabilidade condicional de Bayes

$$P(A|B) = P(A \cap B)/P(B)$$

$$P(B|A) = P(B \cap A)/P(A)$$

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

Classificador Bayesiano

Podemos então remodelar o problema da seguinte forma

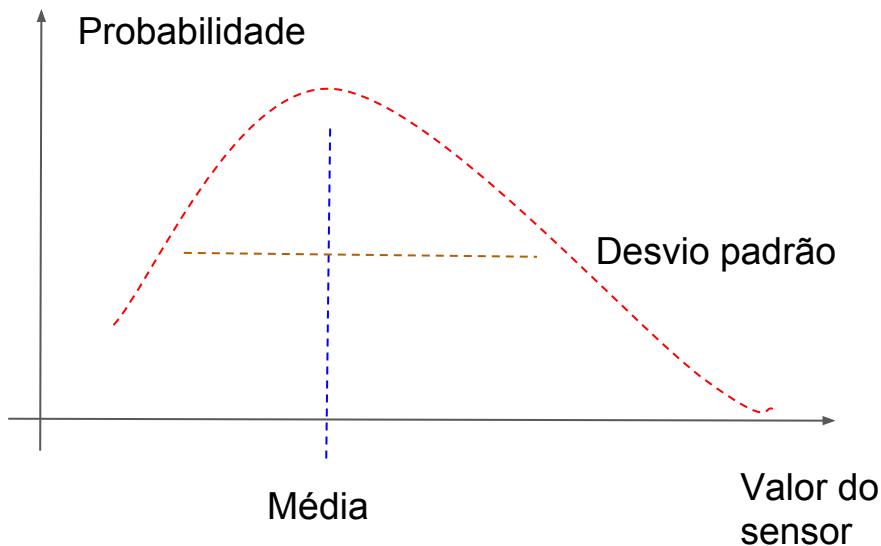
$$P[\text{Salmão} \mid \text{Sensor} = x] = P[\text{Salmão}] * P[\text{Sensor} = x \mid \text{Salmão}] / P[\text{Sensor} = x]$$

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k) p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

Na prática, como o termo $p(x)$ é constante para todas as classes, precisamos apenas do termo $p(C_k) p(\mathbf{x} \mid C_k)$

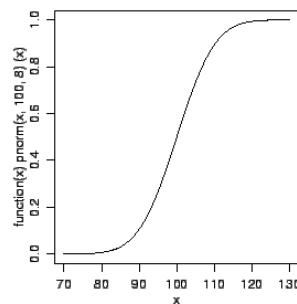
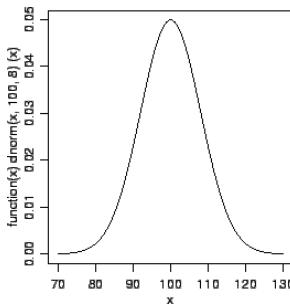
Classificador Bayesiano

Geralmente $P[\text{Sensor} = x]$ e $P[\text{Sensor} = x | \text{Salmão}]$ são descritas por uma função densidade de probabilidade



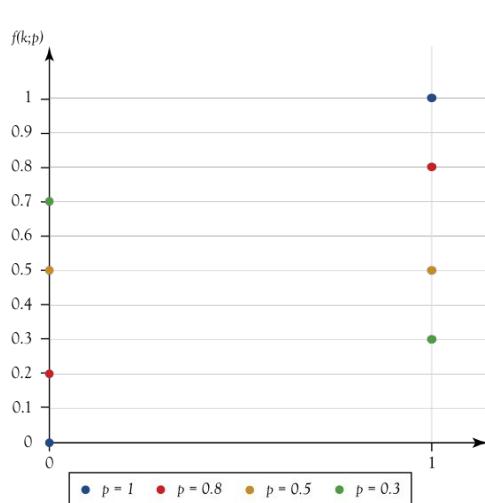
Classificador Bayesiano

Para estimar os parâmetros da função distribuição de probabilidade, o classificador precisa assumir uma determinada forma da distribuição



$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(x - \mu)^2\right\}$$

Distribuição normal



Distribuição de Bernoulli

Classificador Bayesiano

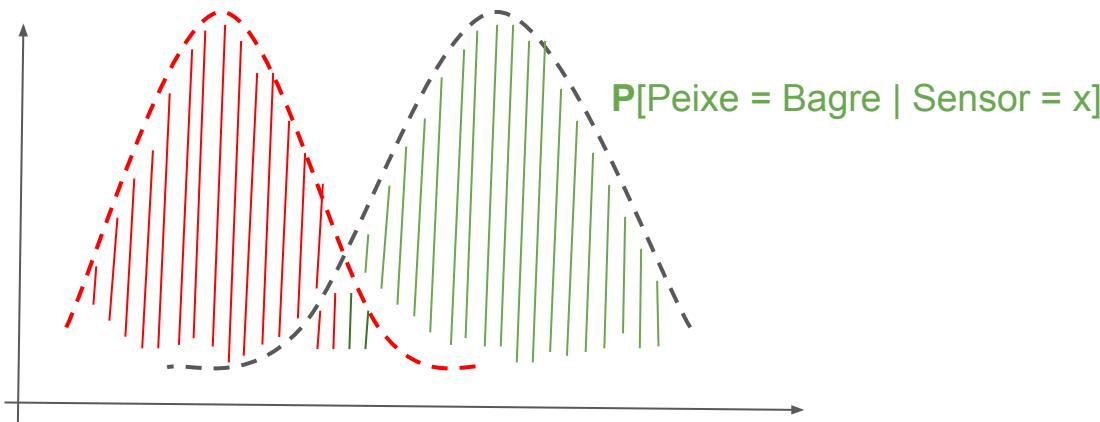
Com base nas funções distribuições de probabilidade, o classificador pode fazer a sua decisão

$$P[\text{Peixe} = \text{Salmão} | \text{Sensor} = x]$$

 Região de escolha - Salmão

$$P[\text{Peixe} = \text{Bagre} | \text{Sensor} = x]$$

 Região de escolha - Bagre



Classificador Bayesiano

Além disso, se houver mais de uma feature, é necessário estimar como elas interagem entre si

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)p(x_2, \dots, x_n, C_k) \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)p(x_2 \mid x_3, \dots, x_n, C_k)p(x_3, \dots, x_n, C_k) \\ &= \dots \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)p(x_2 \mid x_3, \dots, x_n, C_k) \dots p(x_{n-1} \mid x_n, C_k)p(x_n \mid C_k)p(C_k) \end{aligned}$$

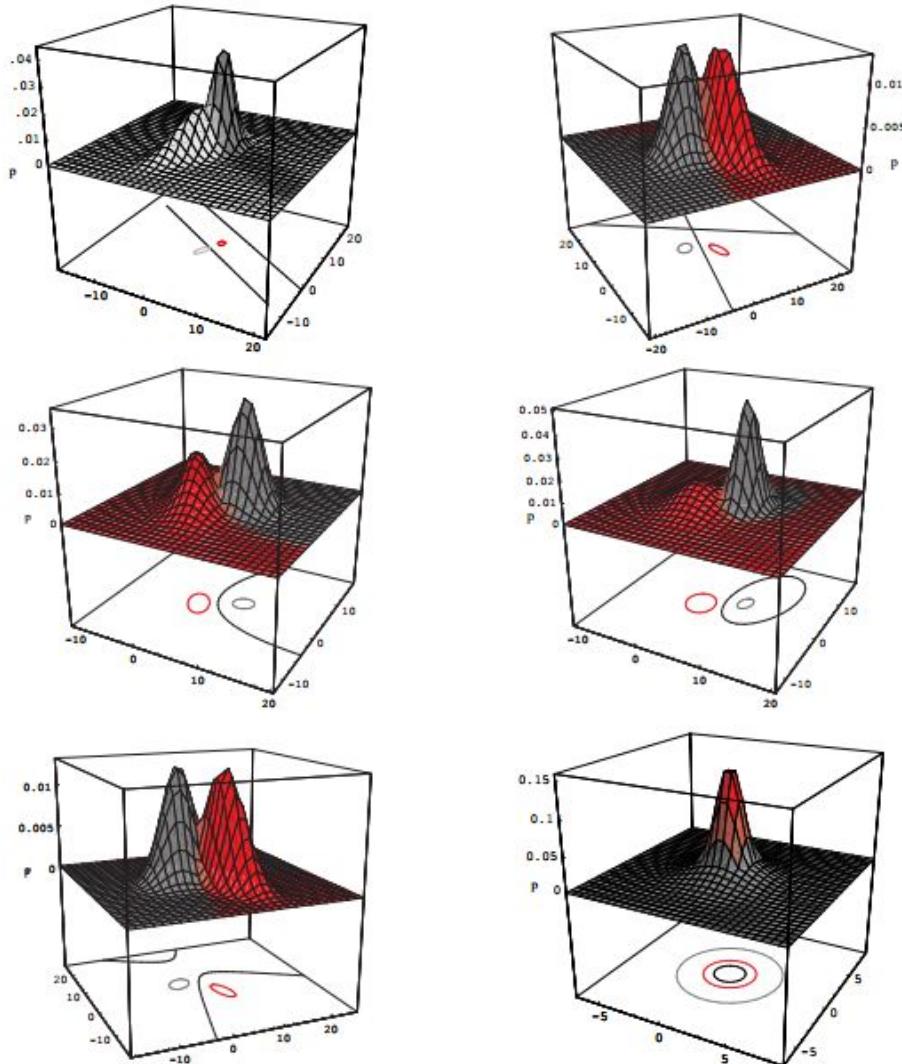
O chamado Naive Bayes assume que as features são independentes

$$\begin{aligned} p(C_k \mid x_1, \dots, x_n) &\propto p(C_k, x_1, \dots, x_n) \\ &\propto p(C_k) p(x_1 \mid C_k) p(x_2 \mid C_k) p(x_3 \mid C_k) \dots \\ &\propto p(C_k) \prod_{i=1}^n p(x_i \mid C_k). \end{aligned}$$

Classificador Bayesiano

Fronteira de decisão

O classificador bayesiano pode gerar diversas fronteiras de decisão dependendo de como são as distribuições dos dados.



Classificador Bayesiano

Vantagens

Flexibilidade da fronteira de decisão (menos que o Knn)

Altamente escalável com o aumento da base de dados

Pode fazer previsões probabilísticas

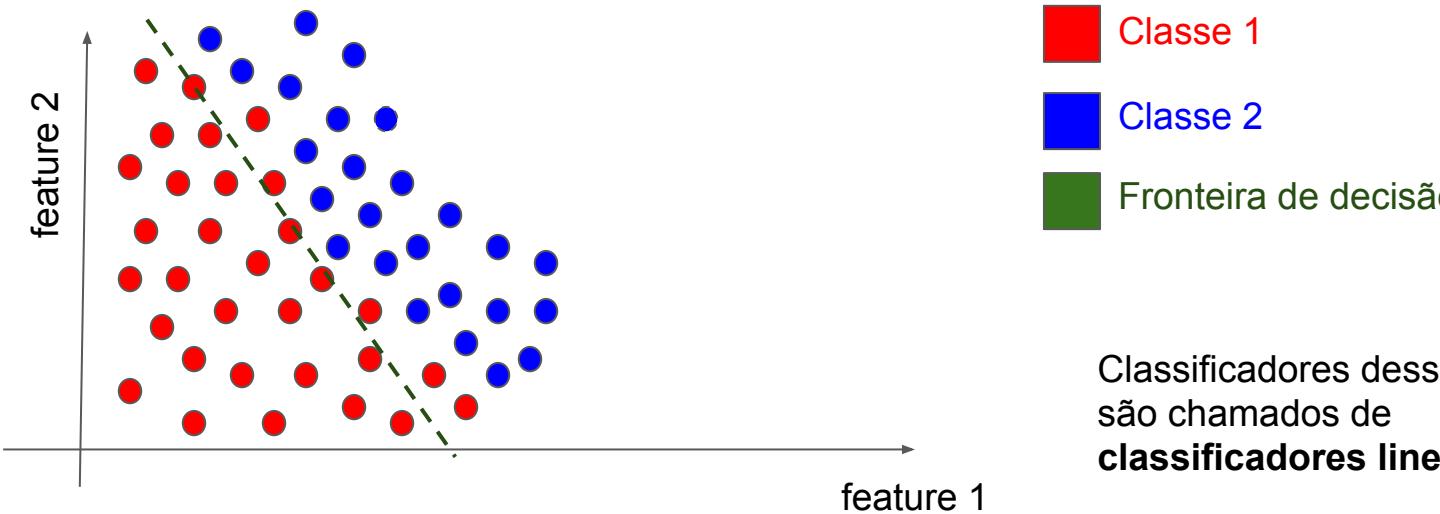
Desvantagens

Relevância na estimação probabilística

Relevância da hipótese da distribuição dos dados

Regressão logística

Um classificador de regressão logística é um classificador binário que tenta separar o espaço através de um hiperplano no espaço das features



Regressão logística

Na vida real, problemas onde a fronteira de decisão é uma reta ou hiperplano são raros, no entanto, prova-se que quanto mais features, maior a probabilidade da fronteira de separação ser linear

Como veremos em breve,
problemas de NLP possuem alta dimensionalidade

 Download full text in PDF Export ▾



IFAC-PapersOnLine

Volume 49, Issue 24, 2016, Pages 64-69



The Blessing of Dimensionality: Separation Theorems in the Thermodynamic Limit *

Alexander N. Gorban *✉, Ivan Yu. Tyukin **✉, Ilya Romanenko ***✉

 [Show more](#)

<https://doi.org/10.1016/j.ifacol.2016.10.755>

[Get rights and content](#)

Abstract:

We consider and analyze properties of large sets of randomly selected (i.i.d.) points in high dimensional spaces. In particular, we consider the problem of whether a single data point that is randomly chosen from a finite set of points can be separated from the rest of the data set by a linear hyperplane. We formulate and prove stochastic

Regressão logística

Em problemas de regressão **linear**, consideramos que a relação entre a entrada e a saída é da seguinte forma

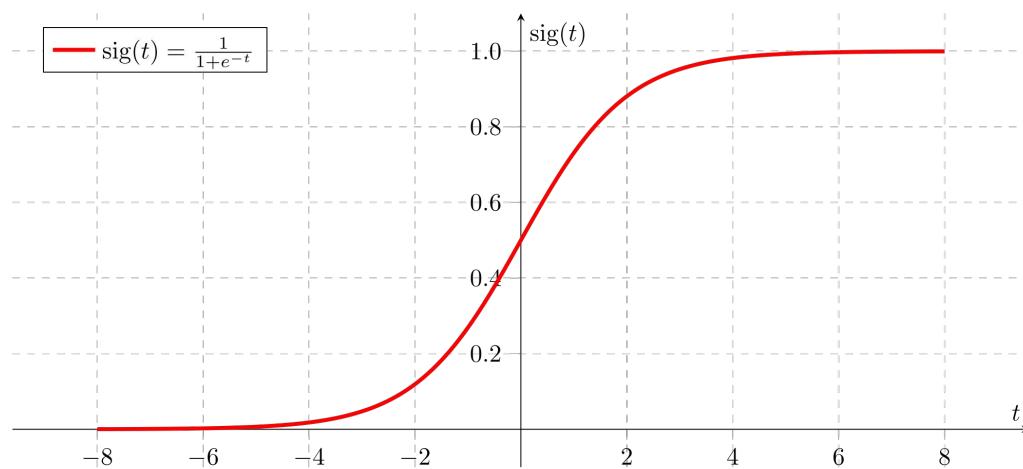
$$\text{Saída} = \text{coef}_1 * (\text{feature}_1) + \text{coef}_2 * (\text{feature}_2) + \text{coef}_3 * (\text{feature}_3) + \dots$$

ou seja

$$\text{Saída} = \sum_i^x \text{coef}_i * (\text{feature}_i)$$

Regressão logística

A diferença é que na regressão logística, aplicamos um *threshold* para realizar a classificação



Por que não uma função degrau? o motivo ficará claro mais tarde

Regressão logística

Função de custo

Na regressão **linear** podemos usar o erro quadrático médio como função de custo

$$Custo = \left(\frac{1}{n} \right) \sum_i (saída_i - saída_{esperada_i})^2$$

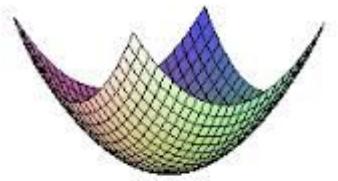
Na regressão logística, utilizamos uma função de custo chamada cross entropy

$$Custo = \left(\frac{1}{n} \right) \sum_i -saída_i * \log(saída_{esperada_i}) - (1 - saída_i) * \log(1 - saída_{esperada_i})$$

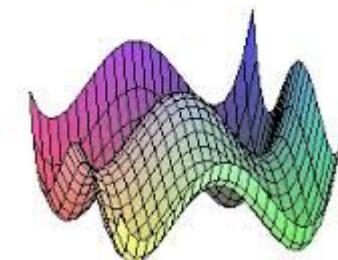
Regressão logística

Função de custo

A função cross entropia tem um único mínimo no problema de classificação



Função cross entropia



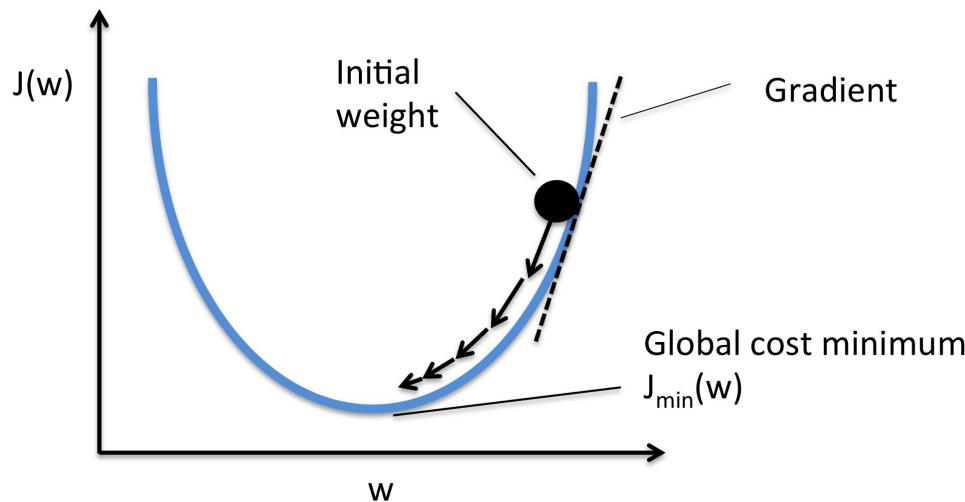
Função MSE em problemas de classificação

Isto significa que, em uma abordagem simplista, é **garantido** que atingiremos o **mínimo global** da função de custo de uma regressão logística

Regressão logística

Atualização de pesos

Como atualizar os pesos de um classificador regressão logística



Taxa de aprendizado

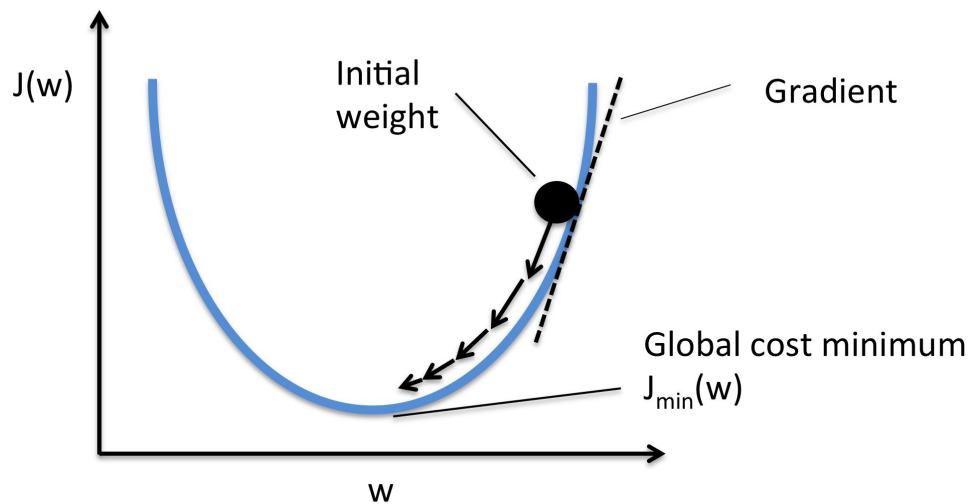
$$\frac{\partial}{\partial x} coef_i := coef_i - \alpha \frac{\partial Custo}{\partial coef}$$

Este é o método **Gradient Descent**, um dos mais utilizados para a atualização dos pesos

Regressão logística

Atualização de pesos

Como atualizar os pesos de um classificador regressão logística



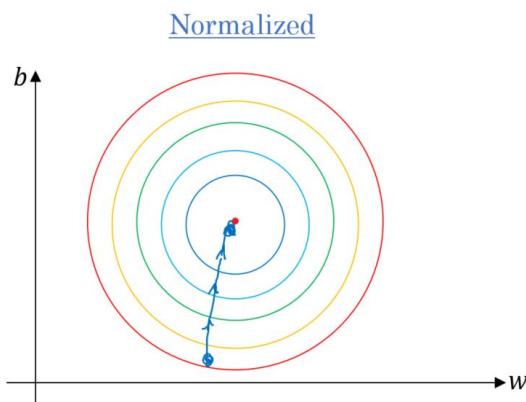
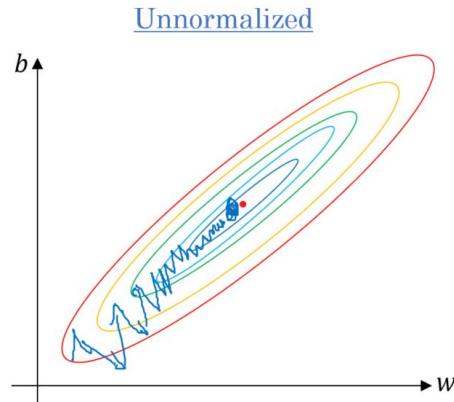
$$\frac{\partial}{\partial x} coe f_i := coe f_i - \alpha \frac{\partial Custo}{\partial coe f}$$

Por isso utilizamos a função sigmóide em vez da função degrau, para que a derivada exista

Regressão logística

Normalização de features

Todo algoritmo de classificação que utiliza gradiente descendente precisa que as variáveis estejam **normalizadas**, ou seja, a ordem de grandeza dos valores não pode ser muito diferente



Regressão logística

Técnicas de normalização

Máximo e mínimo: converte para o intervalo [0, 1]

$$var_{normalizada} = \frac{var_{desnormalizada} - \min(var_{desnormalizada})}{\max(var_{desnormalizada}) - \min(var_{desnormalizada})}$$

Normalização z: converte aproximadamente para o intervalo [-1, 1]

$$var_{normalizada} = \frac{var_{desnormalizada} - \text{média}(var_{desnormalizada})}{\text{std}(var_{desnormalizada})}$$

Codificação de features

No caso de variáveis sem relação de ordem, utilizamos **variáveis dummy**

Tipo de animal: cachorro, gato e coelho

	tipo_animal_0	tipo_animal_1	tipo_animal_2
cachorro	1	0	0
gato	0	1	0
coelho	0	0	1

Codificação de features

No caso de variáveis com relação de ordem, codificar como um inteiro e **normalizar**

Regressão logística

Vantagens

Fácil de interpretar os resultados

Possui apenas um mínimo global

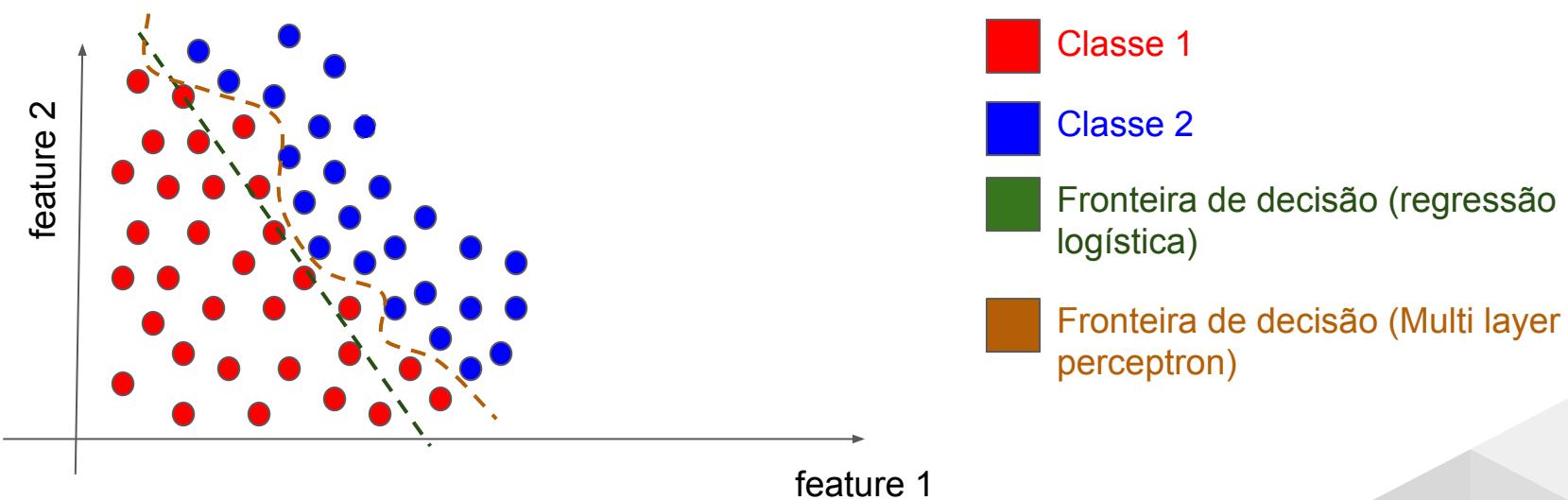
Resolve problemas linearmente separáveis muito bem

Desvantagens

Resolve apenas problemas separáveis linearmente

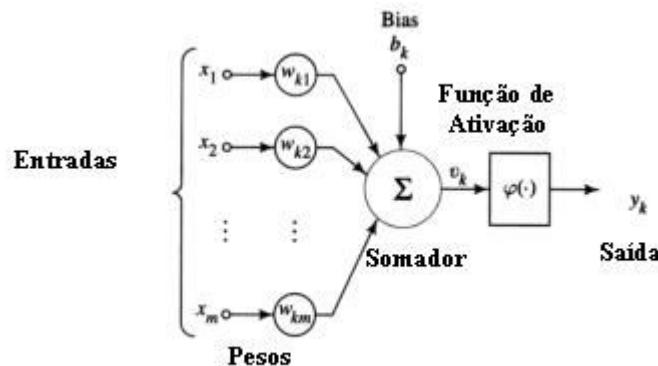
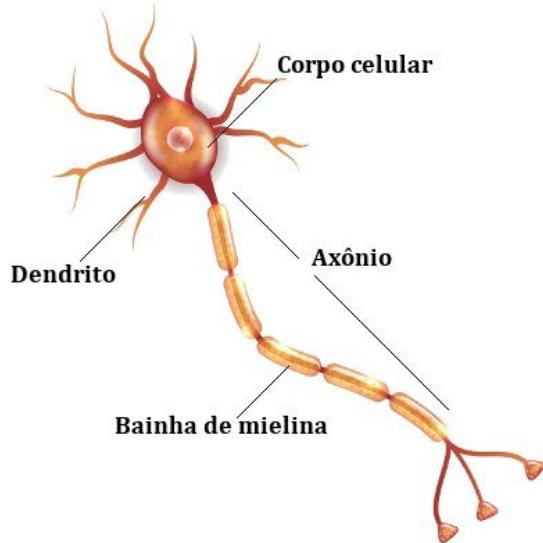
Multi Layer perceptron (Redes neurais shallow)

Este classificador consegue aprender fronteiras de decisão mais complexas do que hiperplanos



Multi Layer perceptron (Redes neurais shallow)

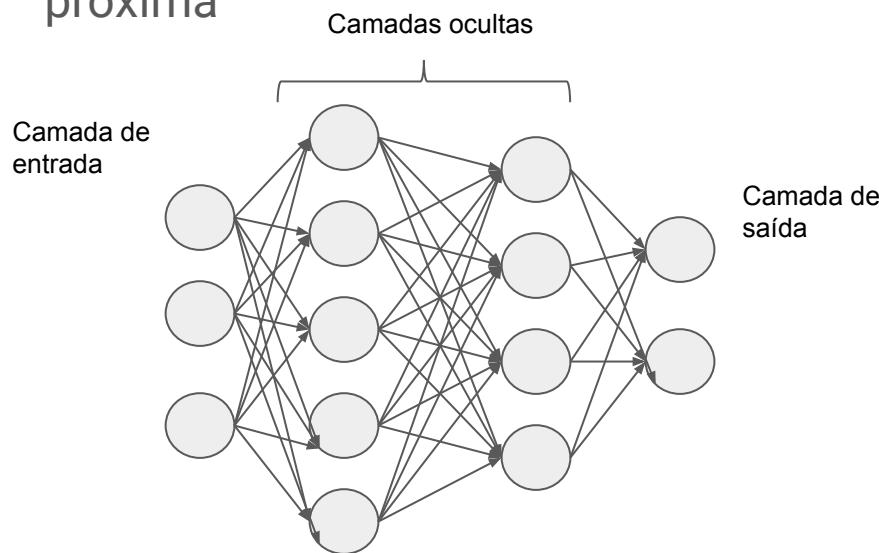
Multi layer perceptrons foram motivados pelo funcionamento do neurônio humano



Cada neurônio é como se fosse uma mini regressão logística

Multi Layer perceptron (Redes neurais shallow)

Em uma rede neural, neurônios são arranjados em camadas de forma que cada camada é totalmente conectada com a próxima



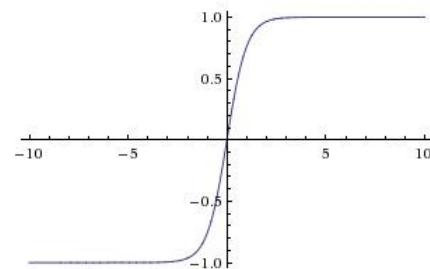
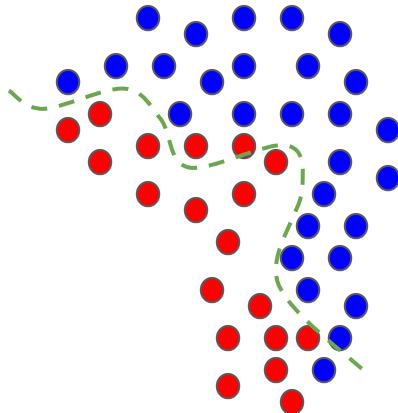
O número de neurônios da camada de entrada é o **número de features**

O número de neurônios da camada de saída é o **número de classes**

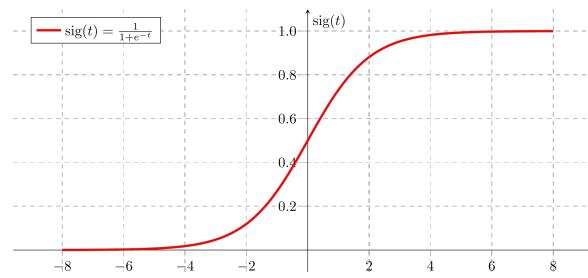
Não existe regra para o número e o tamanho das camadas ocultas, porém + camadas = maior flexibilidade da fronteira de decisão

Multi Layer perceptron (Redes neurais shallow)

A não linearidade é causada pela função de ativação utilizada



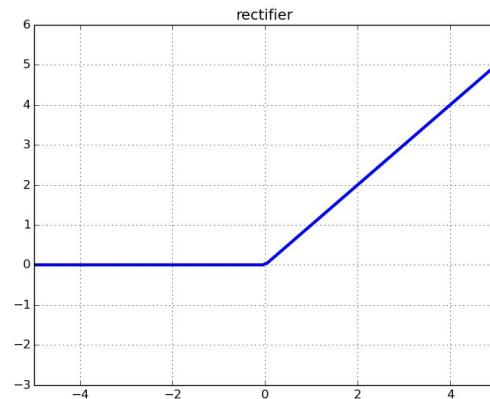
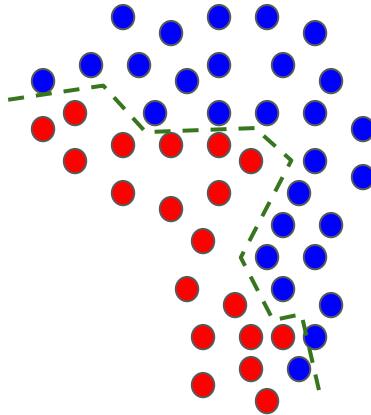
tangente
hiperbólica



sigmoide

Multi Layer perceptron (Redes neurais shallow)

A não linearidade é causada pela função de ativação utilizada



Relu

Multi Layer perceptron (Redes neurais shallow)

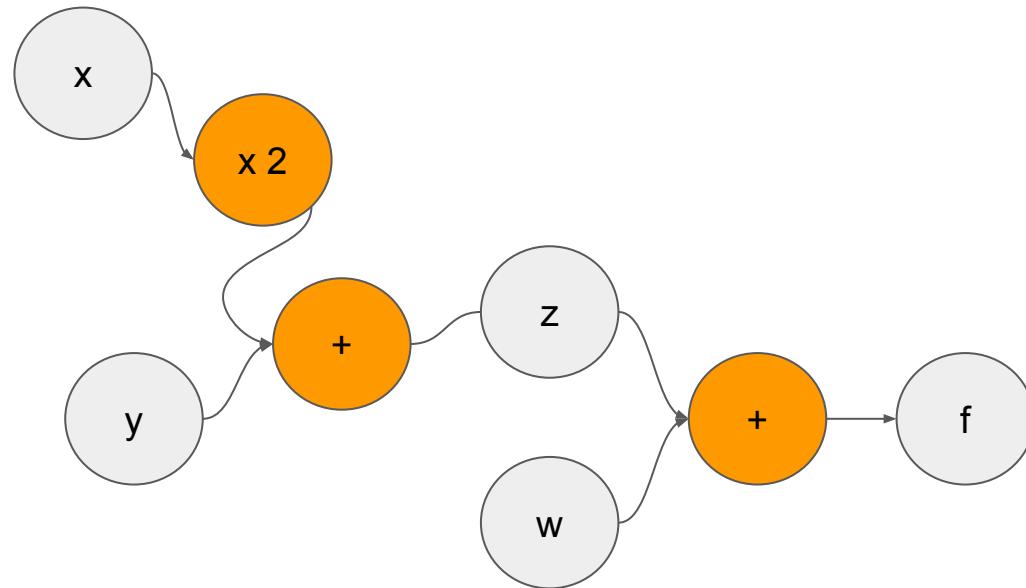
Atualização dos pesos

Também ocorre através de Gradient descent

$$\frac{\partial}{\partial x} \text{coef}_i := \text{coef}_i - \alpha \frac{\partial \text{Custo}}{\partial \text{coef}}$$

A dificuldade é calcular $\frac{\partial \text{Custo}}{\partial \text{coef}_i}$ para os neurônios das camadas escondidas

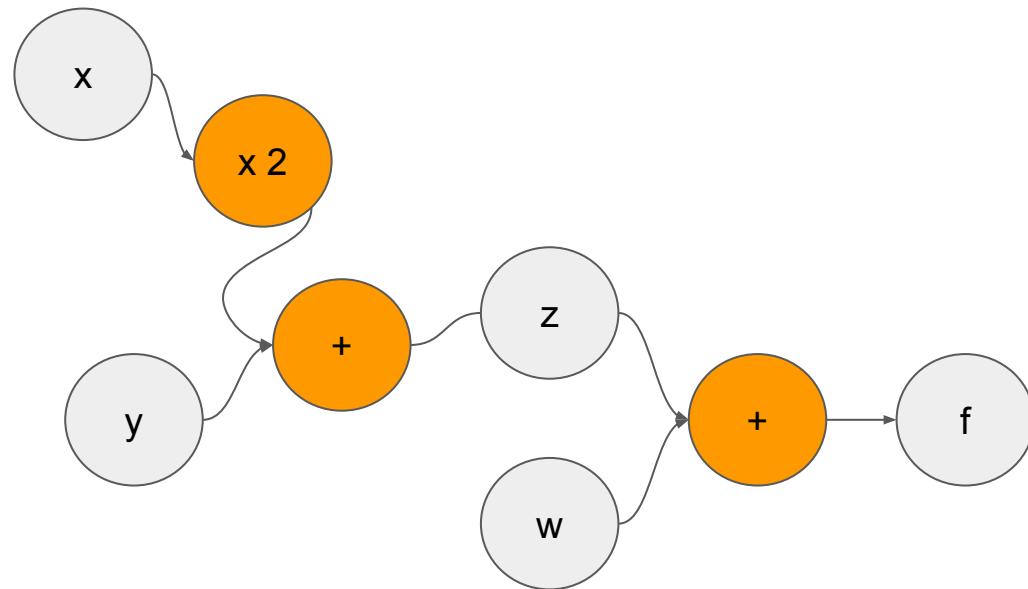
Multi Layer perceptron (Redes neurais shallow)



$$f = z + w$$

$$z = 2x + y$$

Multi Layer perceptron (Redes neurais shallow)



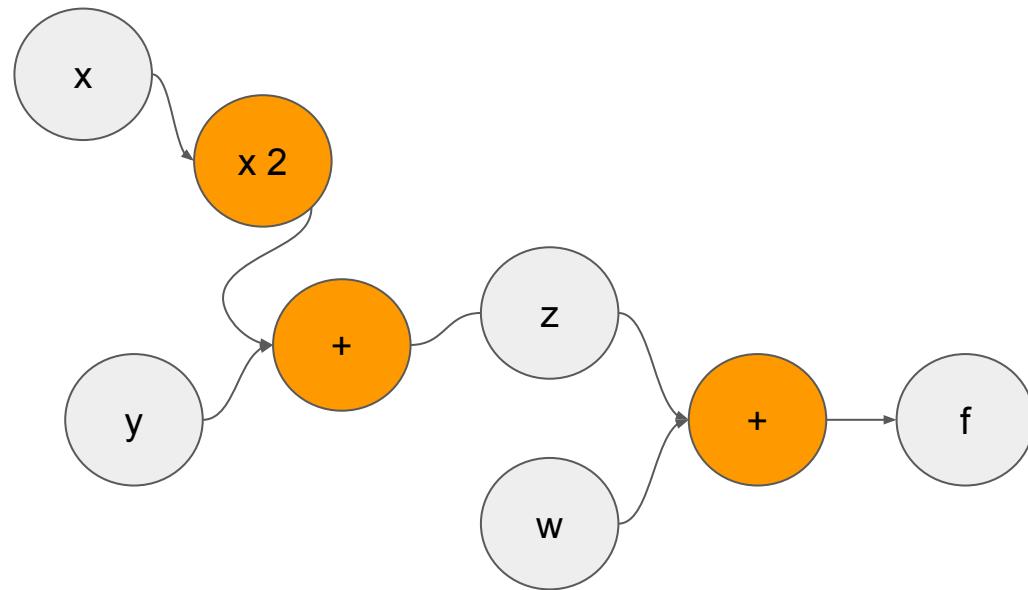
$$f = z + w$$

$$z = 2x + y$$

$$\frac{\partial f}{\partial z} = 1$$

$$\frac{\partial f}{\partial w} = 1$$

Multi Layer perceptron (Redes neurais shallow)



$$f = z + w$$

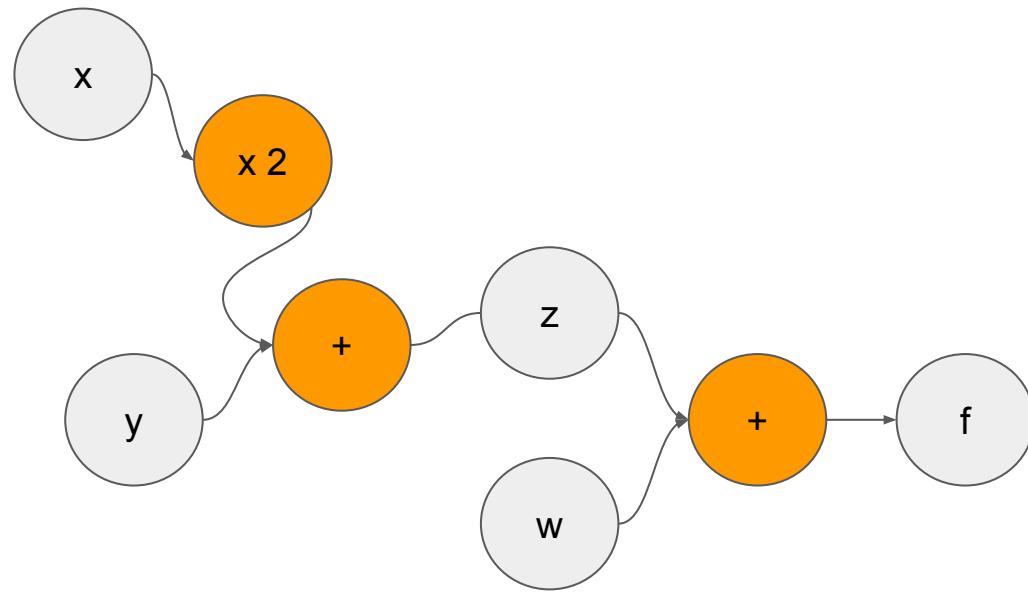
$$z = 2x + y$$

$$\frac{\partial f}{\partial z} = 1$$

$$\frac{\partial f}{\partial w} = 1$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial z} \frac{\partial z}{\partial x}$$

Multi Layer perceptron (Redes neurais shallow)



$$f = z + w$$

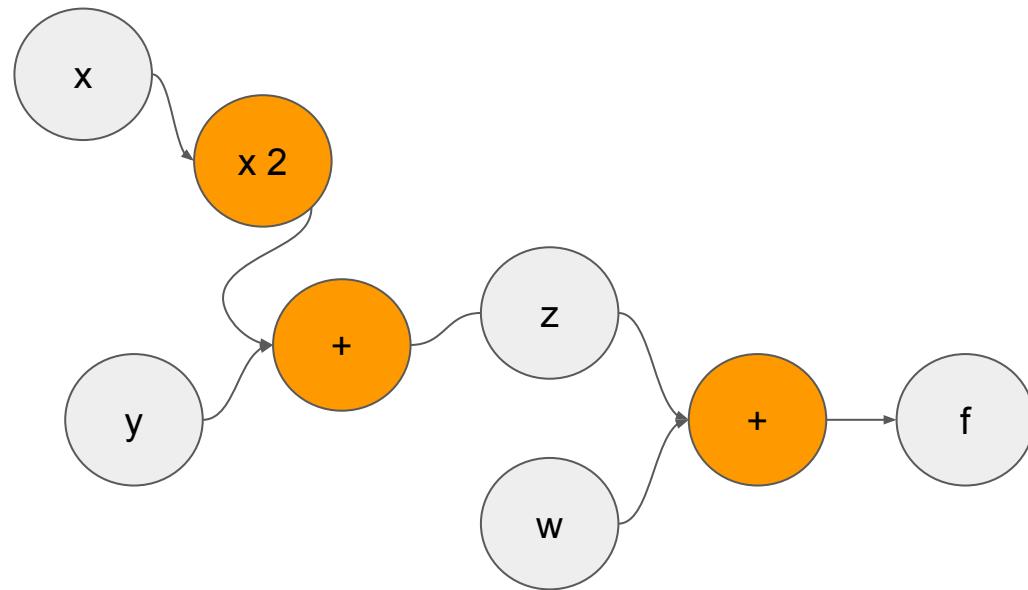
$$z = 2x + y$$

$$\frac{\partial f}{\partial z} = 1 \quad \frac{\partial z}{\partial x} = 2$$

$$\frac{\partial f}{\partial w} = 1$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial z} \frac{\partial z}{\partial x}$$

Multi Layer perceptron (Redes neurais shallow)



$$f = z + w$$

$$z = 2x + y$$

$$\frac{\partial f}{\partial z} = 1 \quad \frac{\partial z}{\partial x} = 2$$

$$\frac{\partial f}{\partial w} = 1$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial z} \frac{\partial z}{\partial x} = 2$$

Multi Layer perceptron (Redes neurais shallow)

Atualização dos pesos

Aplicando este conceito em redes neurais, chegamos às equações do algoritmo Backpropagation

$$\frac{\partial E}{\partial w_{ij}} = o_i \delta_j$$

Não há necessidade de memorizar essas equações

$$\delta_j = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} = \begin{cases} (o_j - t_j)o_j(1 - o_j) & \text{if } j \text{ is an output neuron,} \\ (\sum_{\ell \in L} \delta_\ell w_{j\ell})o_j(1 - o_j) & \text{if } j \text{ is an inner neuron.} \end{cases}$$

Multi Layer perceptron (Redes neurais shallow)

Vantagens

Flexibilidade na fronteira de decisão

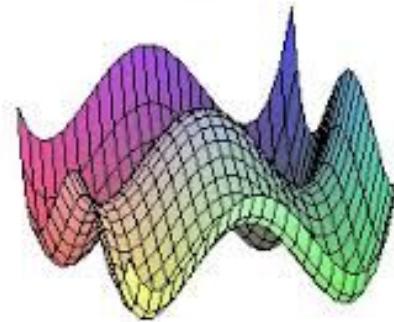
Deu origem ao Deep Learning

Desvantagens

Difícil de interpretar

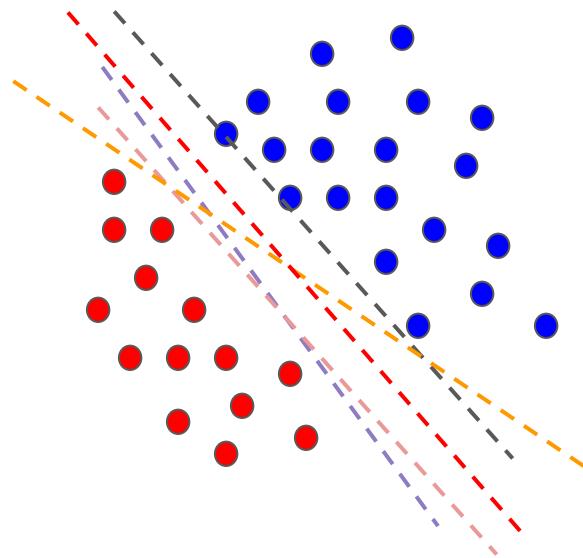
Possui vários mínimos locais

Dificuldade de definir os parâmetros da rede



SVMs (Support Vector Machines)

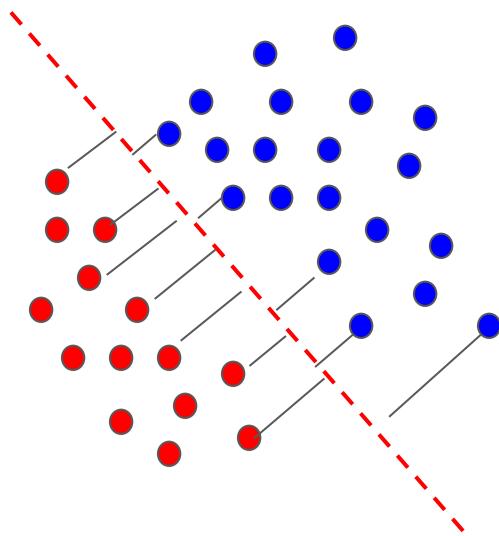
Em sua formulação mais simples, SVMs também são classificadores lineares, porém.....



Neste problema, qual a **melhor reta** para dividir as classes

SVMs (Support Vector Machines)

SVMs garantem a criação de uma reta com o maior distanciamento possível dos pontos experimentais



As retas indicadas na figura
são os chamados **vetores de
suporte**

SVMs (Support Vector Machines)

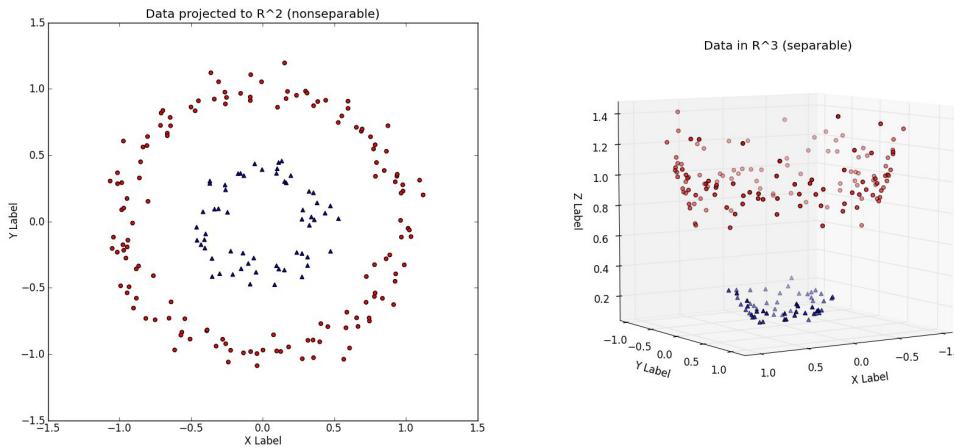
Atualização dos pesos

A atualização de pesos de SVMs envolve minimização quadrática utilizando **multiplicadores de Lagrange**, por simplicidade não mostraremos aqui

SVMs (Support Vector Machines)

Truque do Kernel

Trata-se de um truque matemático para transformar a SVM em um classificador linear

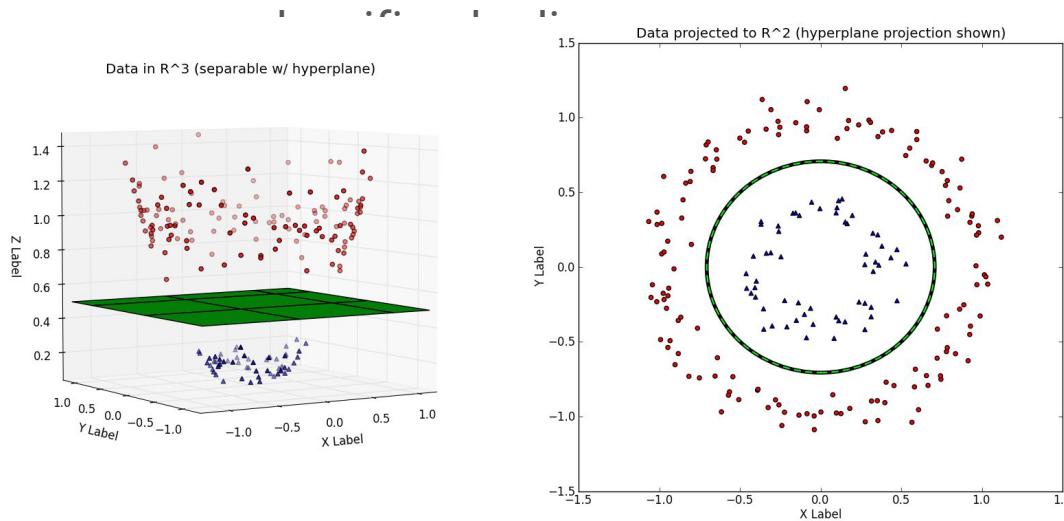


Basicamente, tenta-se representar os dados **não linearmente separáveis** em um novo espaço multidimensional onde eles passam a ser **linearmente separáveis**

SVMs (Support Vector Machines)

Truque do Kernel

Trata-se de um truque matemático para transformar a SVM

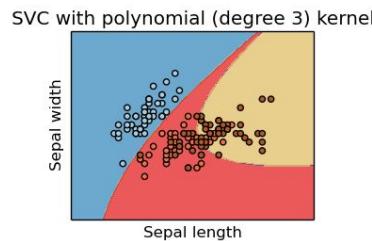
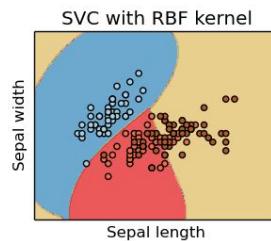
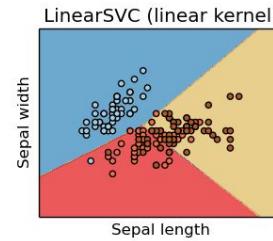
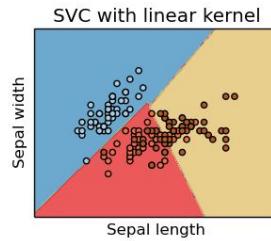


Conforme o mostrado anteriormente, SVMs garantem a **melhor separação linear** no espaço não linear construído

SVMs (Support Vector Machines)

Truque do Kernel

Existem vários kernels canônicos para SVMs linear



SVMs (Support Vector Machines)

Vantagens

Solução de melhor separação para problemas lineares

Boa formulação matemática

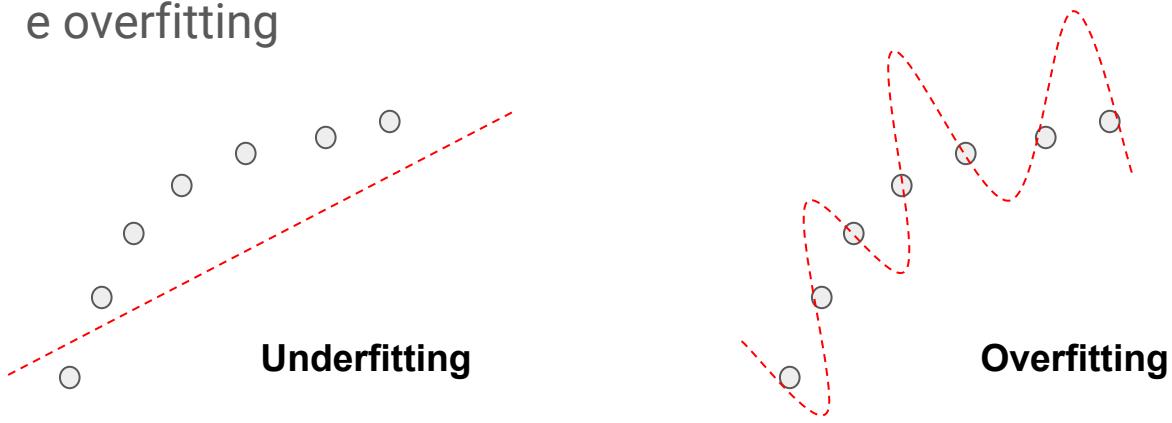
Desvantagens

Dificuldade de encontrar o Kernel certo

Lentas para treinar

Identificando overfitting e underfitting

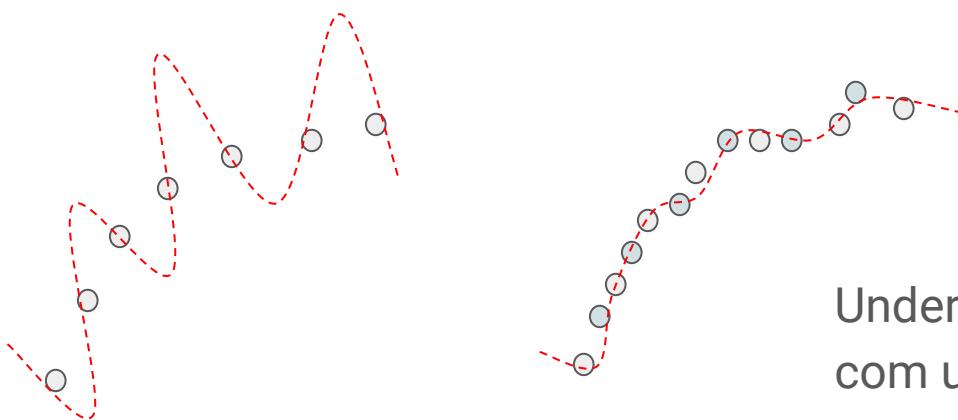
Nas aulas passadas, comentamos o problema de underfitting e overfitting



Basicamente, existe um tradeoff entre um alto poder de representação de um modelo e a sua capacidade de generalizar.

Identificando overfitting e underfitting

Geralmente, quando temos um problema de overfitting, a adição de mais dados de treinamento tende a mitigar o problema

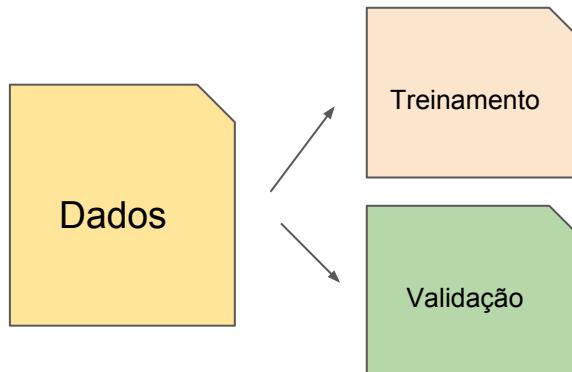


Underfitting, por outro lado, **não** é resolvido com um aumento na quantidade de dados

Identificando overfitting e underfitting

Para identificar em qual situação estamos, é comum a construção de **curvas de aprendizado**

O primeiro passo é a separação dos dados em um conjunto de treinamento e outro de validação

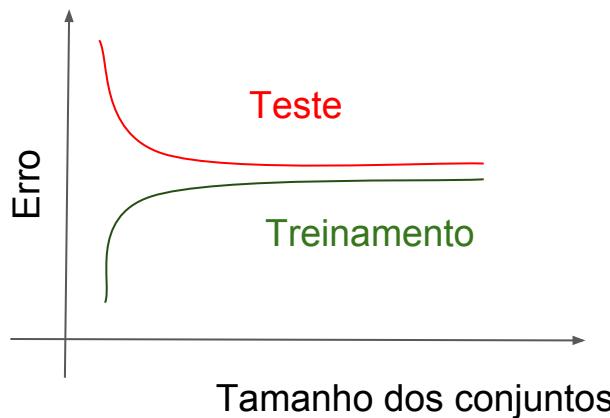


A seguir, controlaremos aos poucos o tamanho dos conjuntos de treinamento e validação

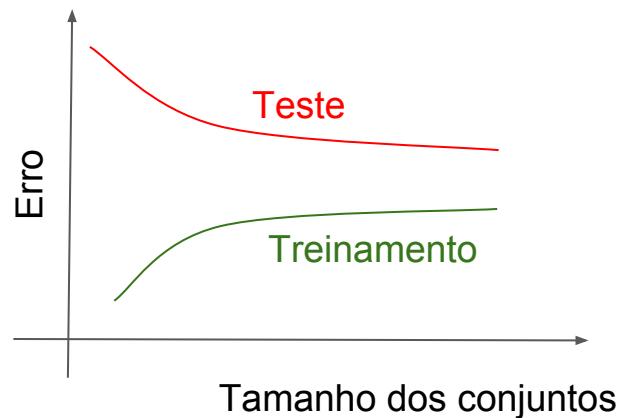
Identificando overfitting e underfitting

Para cada um dos tamanhos dos conjuntos de treinamento e validação, plotamos o erro do classificador em função do tamanho do conjunto. Temos dois casos principais

Underfitting



Overfitting



Combinação de classificadores

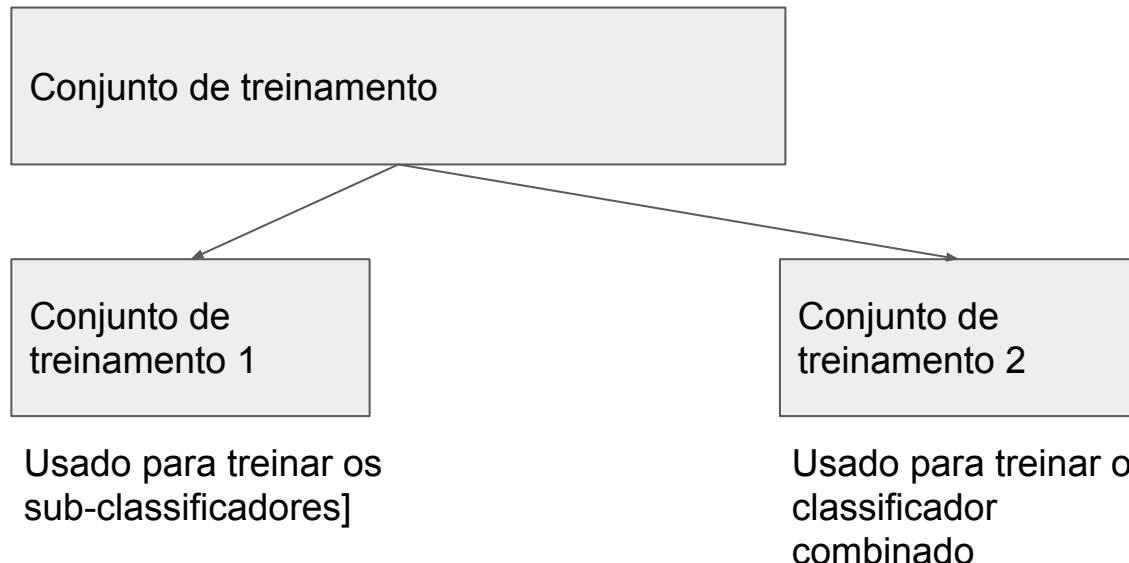
Múltiplos classificadores podem ser combinados. Algumas das técnicas mais simples são:

Voto majoritário: O classificador combinado vai retornar a classificação mais frequente dentre os sub-classificadores

Stacking ou aprendizado no espaço de saídas: A saída dos classificadores anteriores é usada como entrada no treinamento do classificador final

Stacking

No caso do stacking, é importante dividir o conjunto de treinamento em duas partes:



Metodologias mais complexas de combinação

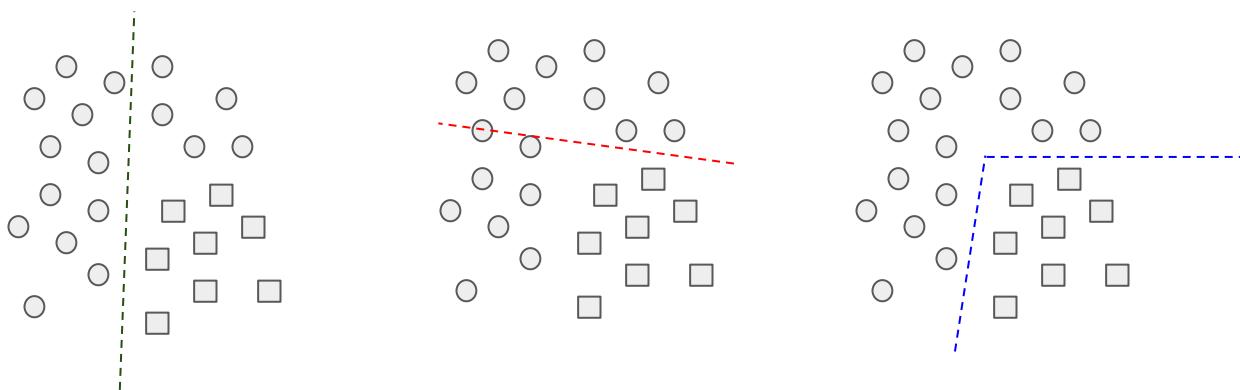
Existem outras duas metodologias de combinação de classificadores que valem a pena ser mencionadas

Adaboost

Bagging

Adaboost

A idéia do adaboost é combinar um conjunto de classificadores **fracos** em um classificador **forte**



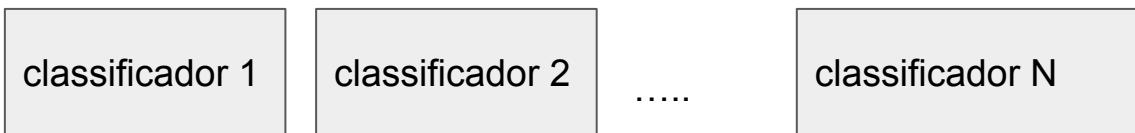
Adaboost

Funcionamento

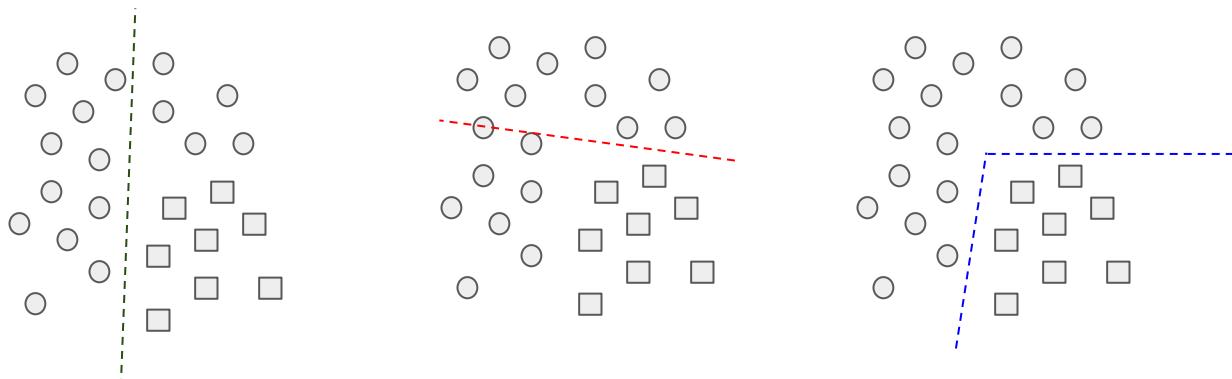
No adaboost, os sub-classificadores são treinados de forma **sequencial**, ou seja, primeiro é treinado um classificador e o mesmo é **testado**. O desempenho deste classificador **influencia** o treinamento do próximo.

Adaboost

Suponha que existam N classificadores



O classificador $N + 1$ é treinado de forma a aprender melhor os exemplos onde os classificadores anteriores foram mal

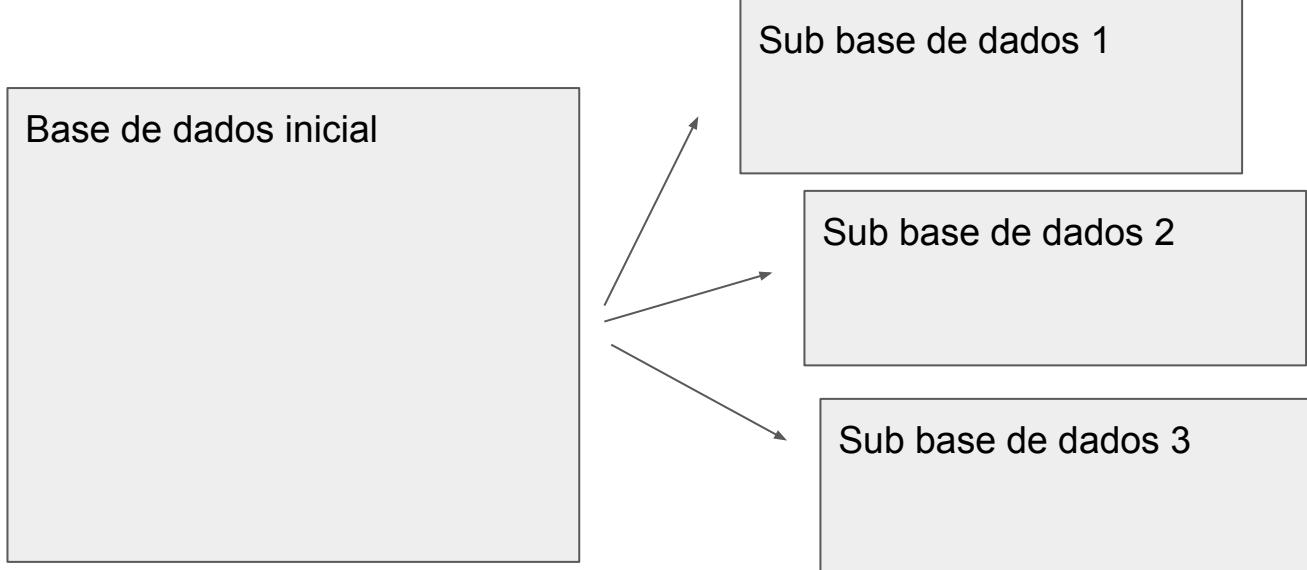


Adaboost

Ao final do treinamento, as saídas dos classificadores são combinadas em uma média ponderada. Os pesos de cada classificador também são aprendidos durante o processo de treinamento.

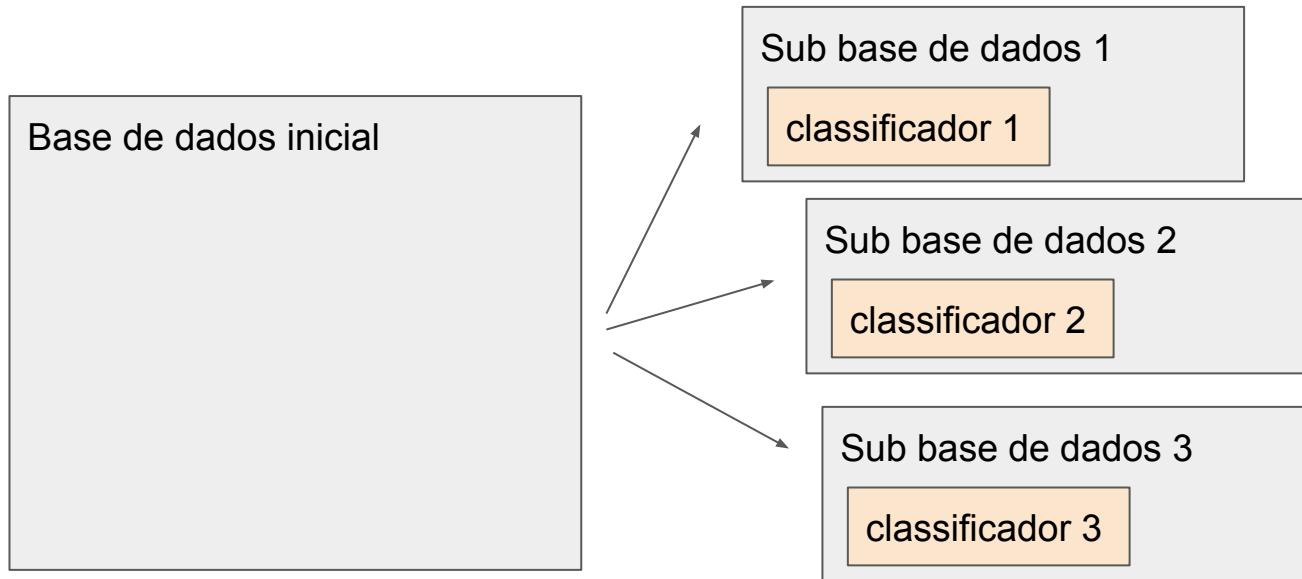
Bagging

O bagging é outra técnica de combinação de classificadores que consiste em dividir a base de dados original em diversas sub-bases



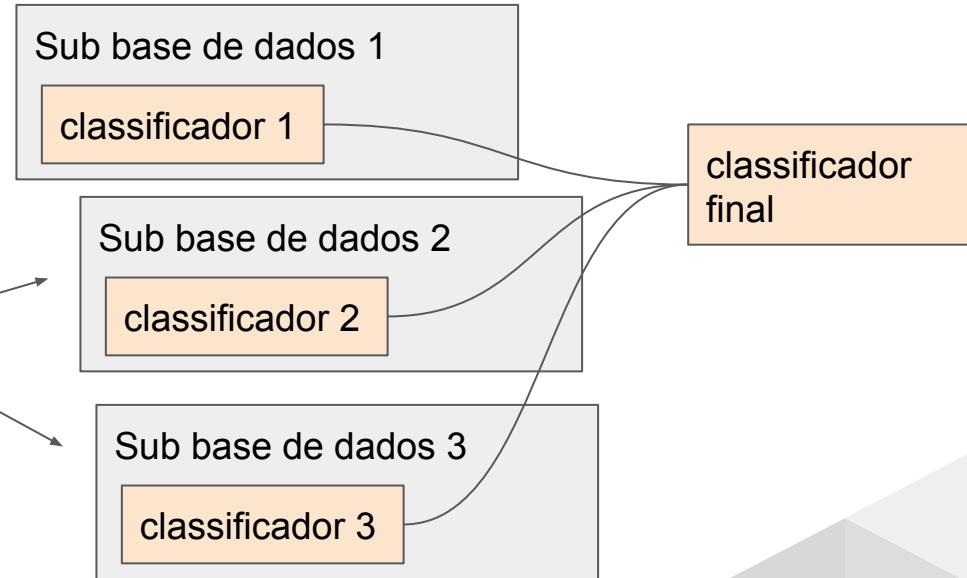
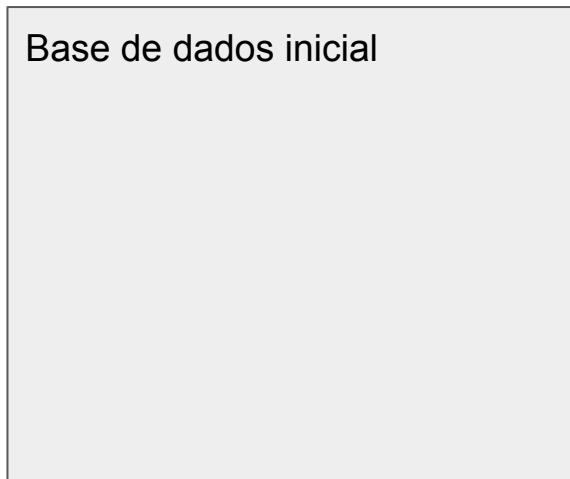
Bagging

Cada uma dessas sub-base de dados darão origem a um novo classificador



Bagging

O classificador final é a combinação desses classificadores, que pode ser feita através do **voto majoritário** ou tirando a média das saídas contínuas dos classificadores



Problemas comuns

Escalabilidade no treinamento

Quando estamos trabalhando com bases de dados muito grandes, surgem problemas no treinamento. A biblioteca sklearn, utiliza uma metodologia de treinamento chamada *treinamento em batch*

Nesta metodologia, **todos os exemplos de treinamento são carregados na memória de uma vez**, a fim de otimizar os cálculos da função de custo

Problemas comuns

Escalabilidade no treinamento

Por exemplo, quando calculamos o erro quadrático médio, o termo

$$\sum_i (saída_i - saída_{esperada_i})^2$$

Pode ser calculado através de uma operação com matrizes, técnica chamada de **vetorização** do cálculo

$$(saída_i - saída_{esperada_i})(saída_i - saída_{esperada_i})^T$$

Bibliotecas como o **numpy** são altamente eficientes com operações matriciais

Problemas comuns

Escalabilidade no treinamento

E se tivermos **Terabytes** de informação? Conseguimos carregar todos os exemplos de treinamento de uma vez na memória?

Alternativa: Treinamento em mini-batches

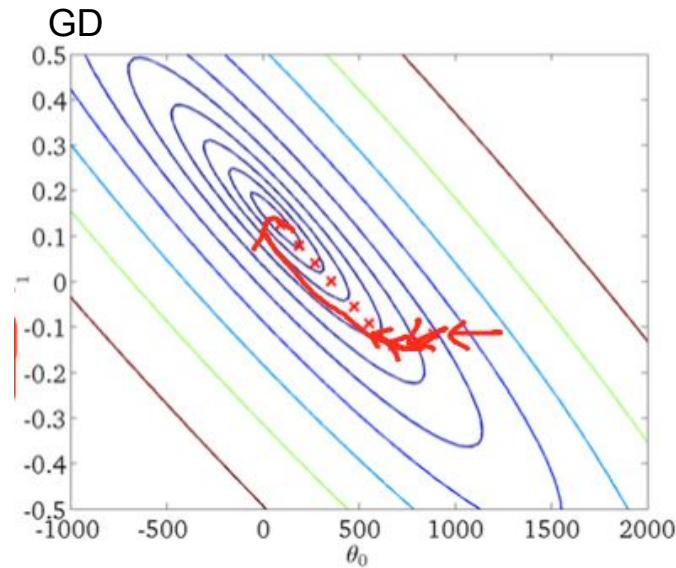
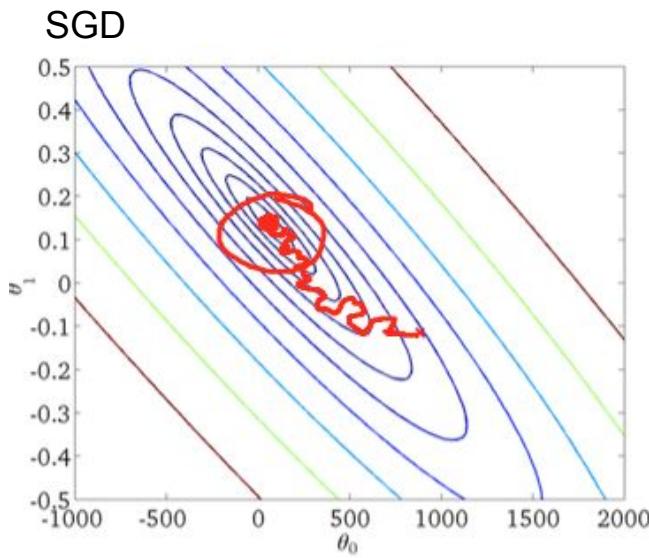
1. Carregamos apenas um subconjunto dos dados por vez e atualizamos os parâmetros do classificador
2. Carregamos o próximo subconjunto
3. Repetimos o processo até que todo os exemplos tenham sido varridos;

Este processo inteiro se chama uma **época**, o treinamento pode ter várias épocas, ou seja, o conjunto de dados pode ser lido várias vezes

Problemas comuns

Escalabilidade no treinamento

Quando o treinamento de mini-batch é aplicado no Gradient descent e os batches são construídos de forma aleatória, o treinamento ganha o nome de *Stochastic Gradient Descent (SGD)*



Problemas comuns

Conjuntos desbalanceados

Muito frequentemente encontramos problemas onde uma das classes acontece muito mais do que a outra. Chamamos estes problemas de **desbalanceados**.

Nesses casos, muitos classificadores tendem a ter um bom resultado na classe mais frequente e um péssimo resultado na classe menos frequente. Como lidar com isso?

Problemas comuns

Conjuntos desbalanceados

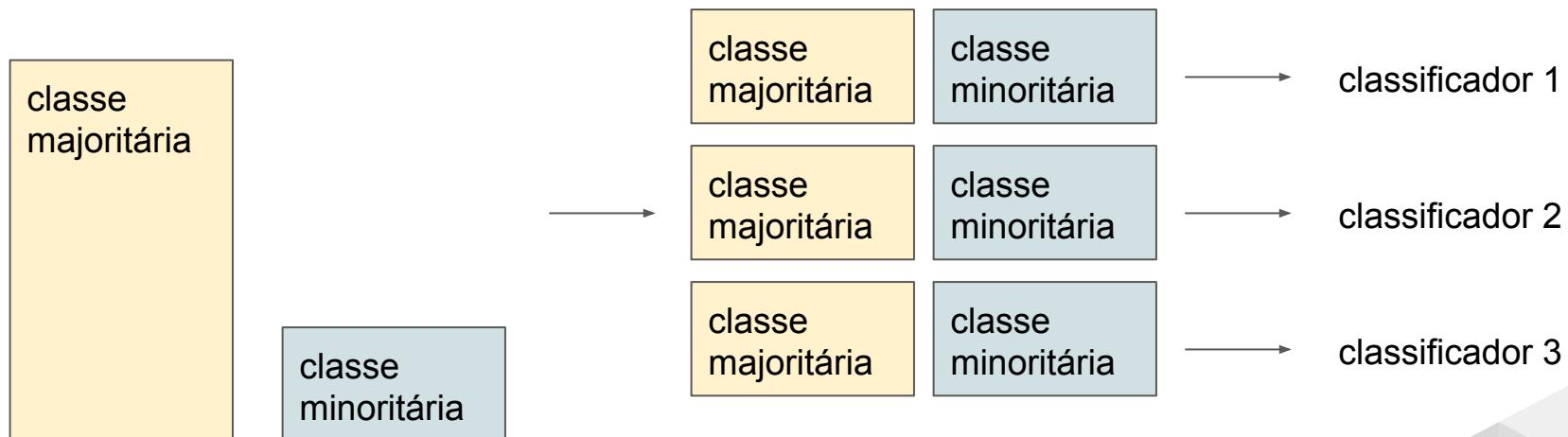
Primeiro passo: Utilizar as métricas certas

Abordagem 1: Undersampling da classe majoritária

Elimine dados da classe majoritária no conjunto de treinamento até que ele fique com a mesma quantidade do classe minoritária. Isto pode ser feito quando a base de dados é grande mesmo para a classe minoritária. **FAÇA ISTO APENAS NO CONJUNTO DE TREINAMENTO.**

Undersampling + ensemble

É possível criar vários conjuntos de treinamento e criar um classificador para cada um. Posteriormente, pode-se combinar os classificadores.

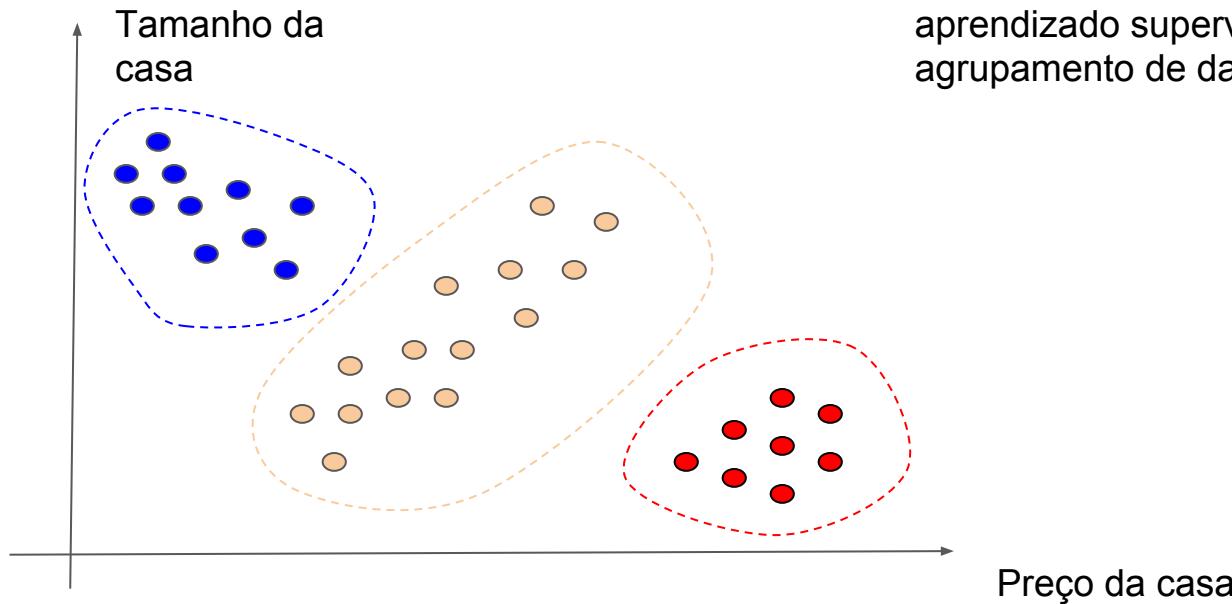


Undersampling com Clustering

Uma forma bastante inteligente de realizar um undersampling na base é fazer um agrupamento dos dados, ou seja, representar um conjunto de registro por um único. Para tal, devemos utilizar técnicas de aprendizado **não supervisionado**.

Clustering e aprendizado não supervisionado

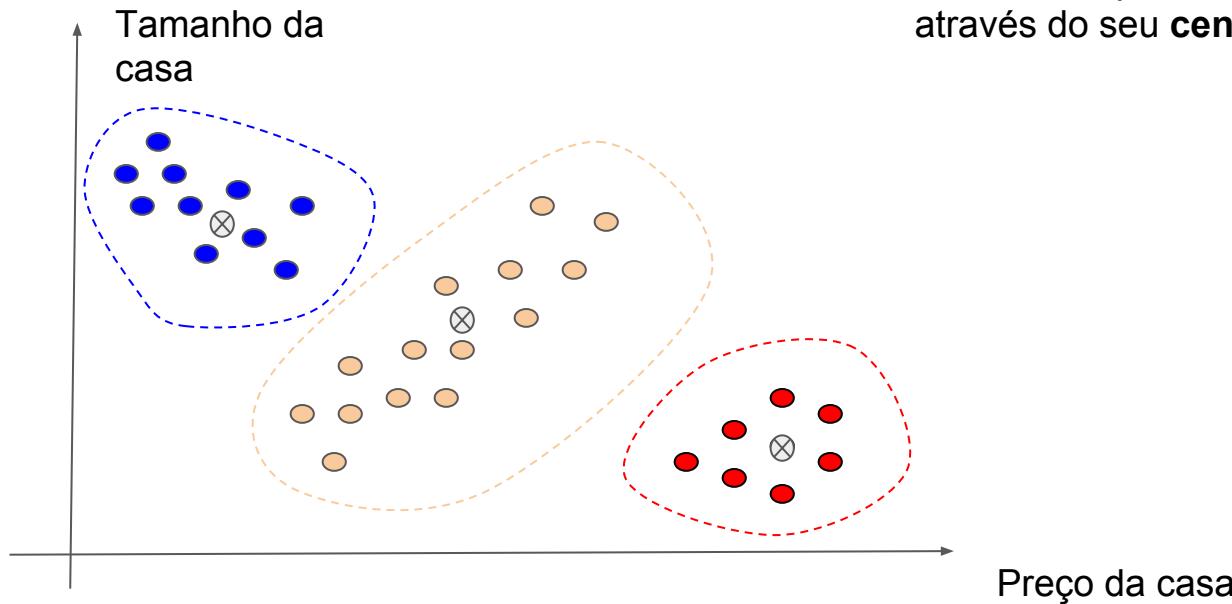
Exemplo



Como dito anteriormente, técnicas de aprendizado supervisionado podem realizar o agrupamento de dados

Clustering e aprendizado não supervisionado

Exemplo



Podemos representar cada um dos grupos através do seu **centróide**

Clustering e aprendizado não supervisionado

Observe então que se desejamos reduzir um conjunto de dados de 1000 elementos para um conjunto de 200 elementos, basta realizar um clustering com 200 centróides.

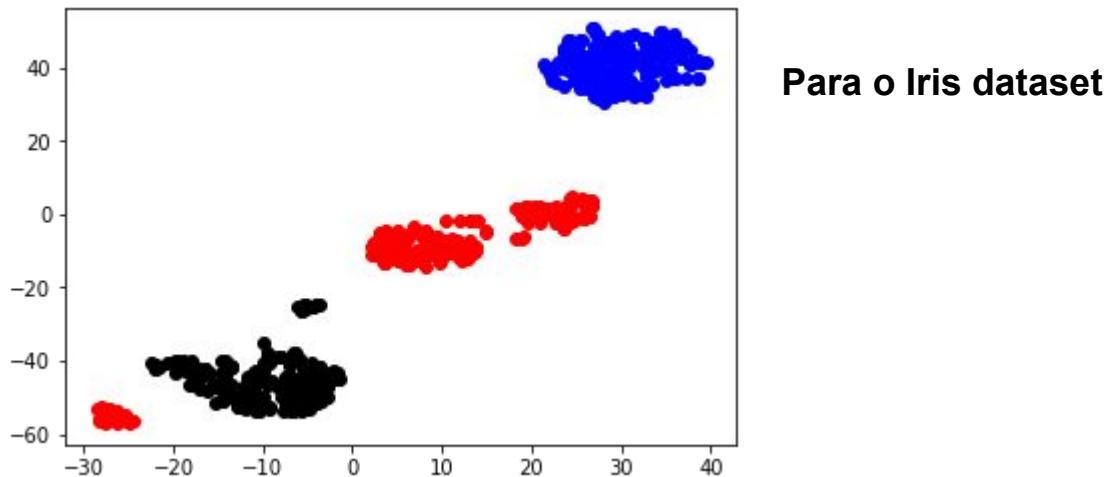
K means

O K means é um dos algoritmos mais simples para a realização de agrupamento de dados. Para utilizar este algoritmo, precisamos que todas as variáveis estejam em sua forma numérica e de uma métrica de distância, tal qual foi feito com o algoritmo **K vizinhos**.

O segundo passo é determinar quantos agrupamentos queremos. Esta determinação pode ser arbitrária (Ex: quando fazemos undersampling) ou pode ser determinada através de um gráfico TSNE.

K means

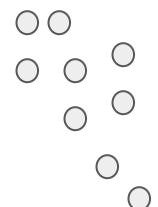
Como podemos observar, para o Iris dataset, podemos utilizar 4 clusters, apesar de somente existirem 3 classes



K means

Funcionamento

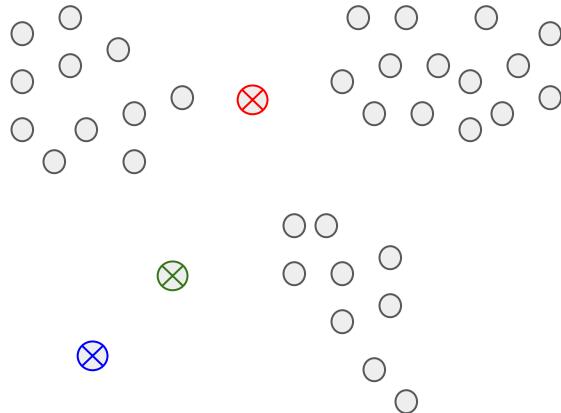
O K means funciona da seguinte forma: Primeiramente, os centróides são inicializados de forma aleatória:



K means

Funcionamento

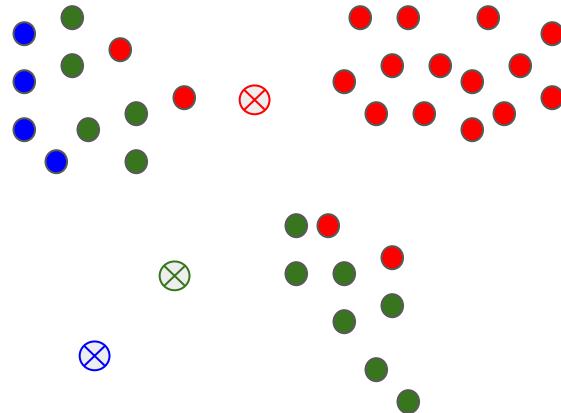
O K means funciona da seguinte forma: Primeiramente, os centróides são inicializados de forma aleatória:



K means

Funcionamento

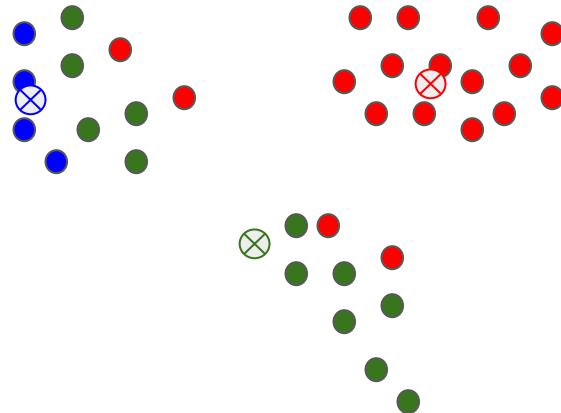
Depois, os pontos são classificados de acordo com qual é o centróide mais próximo



K means

Funcionamento

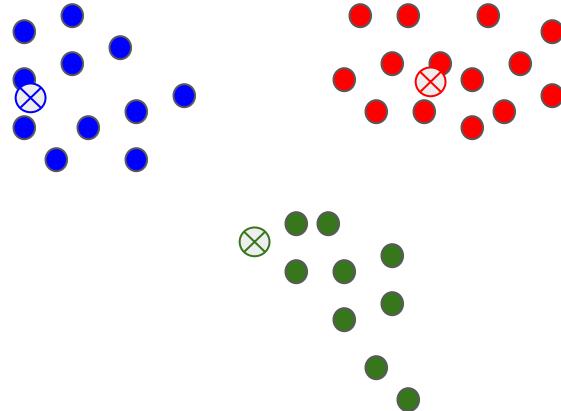
Posteriormente, os centróides são atualizados como sendo o centro dos pontos correspondentes



K means

Funcionamento

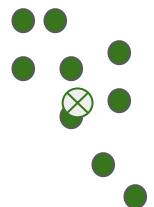
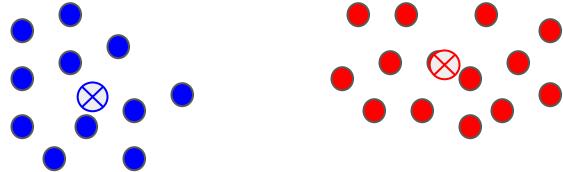
Os pontos são então re-classificados



K means

Funcionamento

Novamente, os centros são atualizados



Este procedimento é repetido por um número fixo de vezes definido pelo usuário ou até que os centros não estejam atualizando

Problemas comuns

Conjuntos desbalanceados

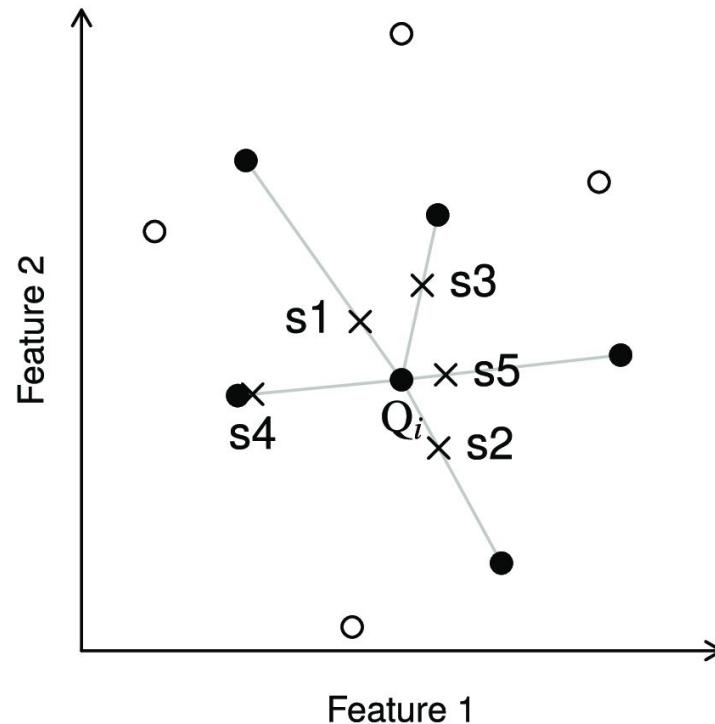
Primeiro passo: Utilizar as métricas certas

Abordagem 2: Oversampling da classe minoritária

Também é possível realizar a repetição ou a criação de novos dados da classe minoritária. **FAÇA ISTO APENAS NO CONJUNTO DE TREINAMENTO.** Uma forma inteligente de se fazer isto é utilizar a técnica **SMOTE**

SMOTE

O SMOTE (Synthetic Minority Oversampling Technique) é uma forma de criar dados artificiais extras a partir da técnica K - vizinhos. Ele toma um ponto aleatório da classe minoritária, calcula os seus K vizinhos e adiciona novos pontos entre esses vizinhos



Problemas comuns

Conjuntos desbalanceados

Primeiro passo: Utilizar as métricas certas

Abordagem 3: Modificar os classificadores

Pode-se também alterar os classificadores para dar mais importância para uma das classes durante o treinamento. Muitos classificadores implementados no sklearn possuem o atributo *class weight* para permitir essa modificação.

Redução de dimensionalidade

Outro problema que frequentemente ocorre em problemas de aprendizado de máquina é um número muito grande de features, o que pode consumir muito poder computacional.

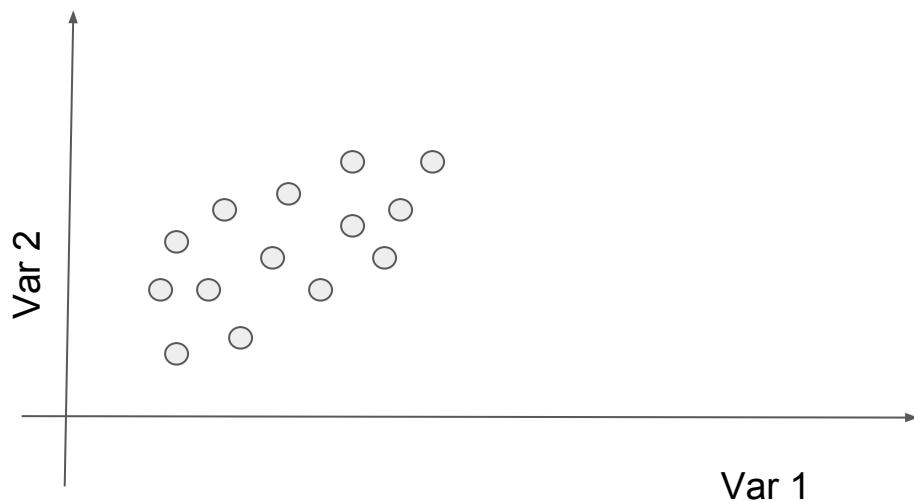
Como podemos reduzir o número de features **sem comprometer** significativamente o modelo?

Redução de dimensionalidade

Uma forma é tentar realizar a filtragem pela métrica ganho de informação, já discutida anteriormente. Outra forma de fazer isto é através de uma técnica conhecida como **PCA**, ou Principal Component Analysis.

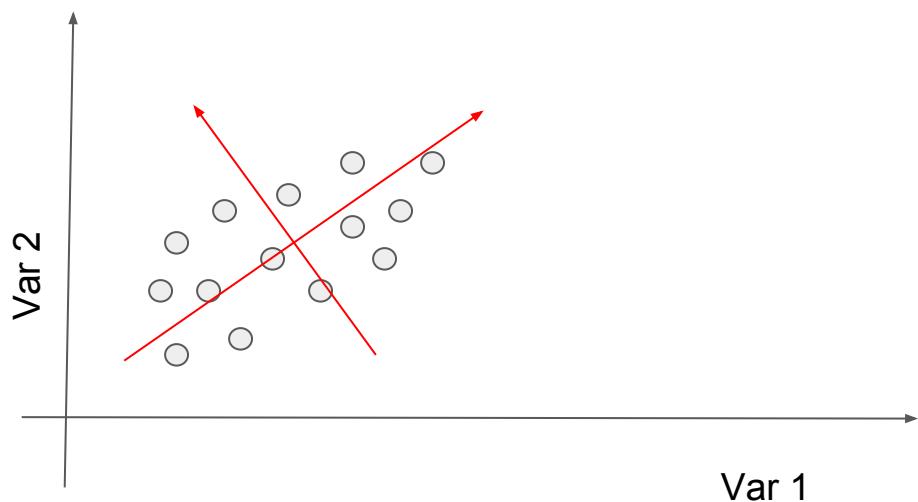
Redução de dimensionalidade

A idéia do PCA é transformar um sistema de coordenadas em um outro sistema de coordenadas onde cada variável é o mais independente da outra o possível



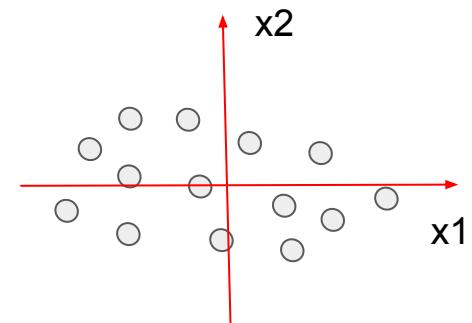
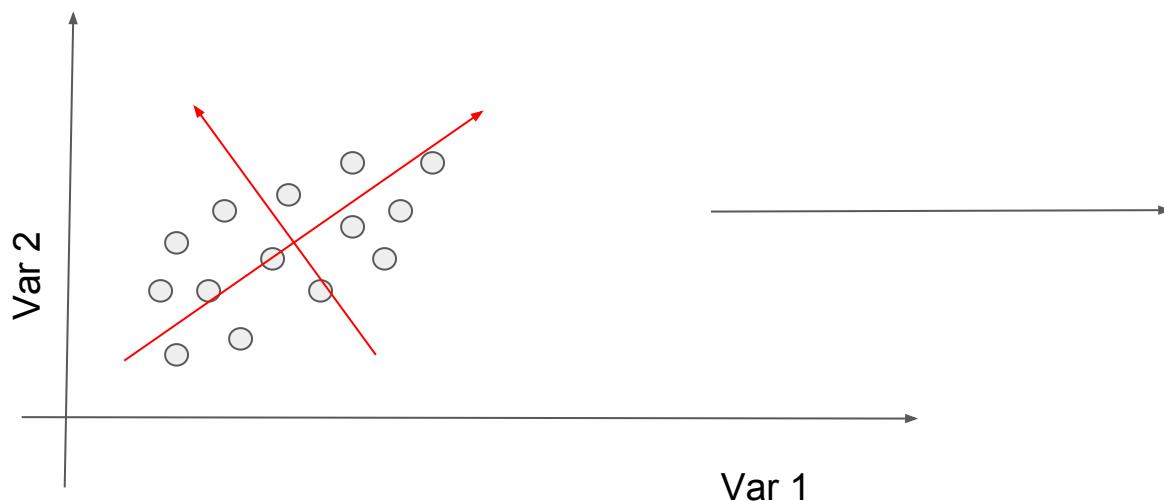
Redução de dimensionalidade

Novo sistema de coordenadas



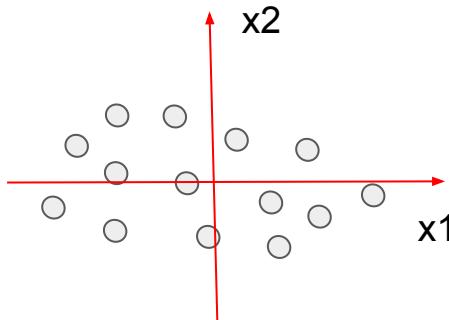
Redução de dimensionalidade

Novo sistema de coordenadas



Redução de dimensionalidade

Neste novo sistema de coordenadas, podemos ordenar as novas variáveis pela variância que elas têm. variáveis com **maior variância** tendem a ter **mais informação**.



Basta então filtrar as n variáveis derivadas com mais variância

Processamento de linguagem natural

Felipe Navarro Balbino Alves

Processamento de linguagem natural

Processamento de linguagem natural nada mais é do que a aplicação de todas as técnicas que discutimos a textos **não estruturados**. Desta forma, discutiremos a partir de agora como realizar codificação de texto em matrizes numéricas e algumas engenharias de features que podem ser realizadas para texto.

Algumas aplicações

Estas são algumas aplicações de processamento de linguagem natural que estudaremos ao longo do curso:

Classificação de texto

Detecção de SPAM, Classificação de sentimento, Análise do resultado de uma ação jurídica

POS tagging

Identificação da função sintática de uma palavra no texto

Reconhecimento de entidade nomeada

Identificação de partes de um texto com um determinado significado, ex: Nomes de pessoas ou empresas

Outras aplicações

Estas são algumas aplicações mais complexas

Tradução

Google Tradutor

Extração de informação

Google agenda

Outras aplicações

Estas são algumas aplicações ainda mais complexas

Conversação de máquina

Google Now. É complexo pois exige ferramentas muito complexas de extração de informação e entendimento de significados

Sumarização e auto resumo

Geração automática de texto

Aplicações híbridas

Algumas aplicações podem gerar texto a partir de mídias mistas, por exemplo, imagens

Describes without errors	Describes with minor errors	Somewhat related to the image	Unrelated to the image
			
<p>A person riding a motorcycle on a dirt road.</p>	<p>Two dogs play in the grass.</p>	<p>A skateboarder does a trick on a ramp.</p>	<p>A dog is jumping to catch a frisbee.</p>
			
<p>A group of young people playing a game of frisbee.</p>	<p>Two hockey players are fighting over the puck.</p>	<p>A little girl in a pink hat is blowing bubbles.</p>	<p>A refrigerator filled with lots of food and drinks.</p>
			
<p>A herd of elephants walking across a dry grass field.</p>	<p>A close up of a cat laying on a couch.</p>	<p>A red motorcycle parked on the side of the road.</p>	<p>A yellow school bus parked in a parking lot.</p>

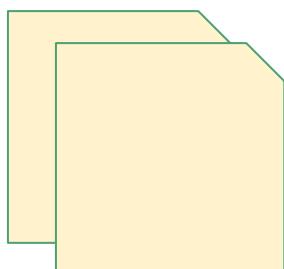
Estado da arte do NLP

Atualmente, as maiores inovações em processamento de linguagem natural estão ligadas com a forma em que as palavras são representadas. As técnicas que estudaremos neste curso tratam palavras como variáveis categóricas

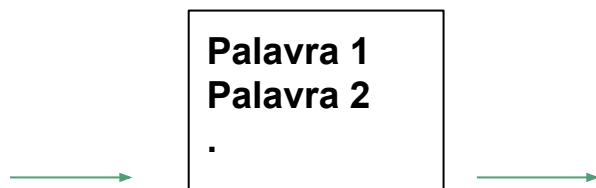
“No desenvolvimento de sistemas de processamento de linguagem natural, as palavras são tratadas como unidades atômicas”

Mikolov. et. al

Estado da arte do NLP



Conjunto de
dados



Conjunto
de n
palavras
distintas

Palavra 1 = [1, 0, , 0]
Palavra 2 = [0, 1, , 0]

.

Palavra k = [0,....., 1, 0...0]

k-1 zeros n-k zeros

Vantagens da representação convencional de palavras

Esparsidade

Consome menos memória

Alguns classificadores off-the-shelf possuem otimizações para matrizes esparsas

Simplicidade

Facilidade em representar documentos (basta somar os vetores das palavras)

Bom para classificação de documentos

Bom desempenho para um volume grande de dados

Desvantagens da representação convencional de palavras

Alta dimensionalidade

Necessita de muitos dados rotulados para atingir um bom desempenho

Perde-se a noção de proximidade semântica entre palavras

Todas as palavras estão nos vértices de um hipercubo de tal forma que as distâncias euclidianas e cosseno são iguais para quaisquer par de palavras

Perde-se a noção de ordem

Codificar N-gramas é uma abordagem, a dimensionalidade aumenta consideravelmente

Estado da arte da representação vetorial

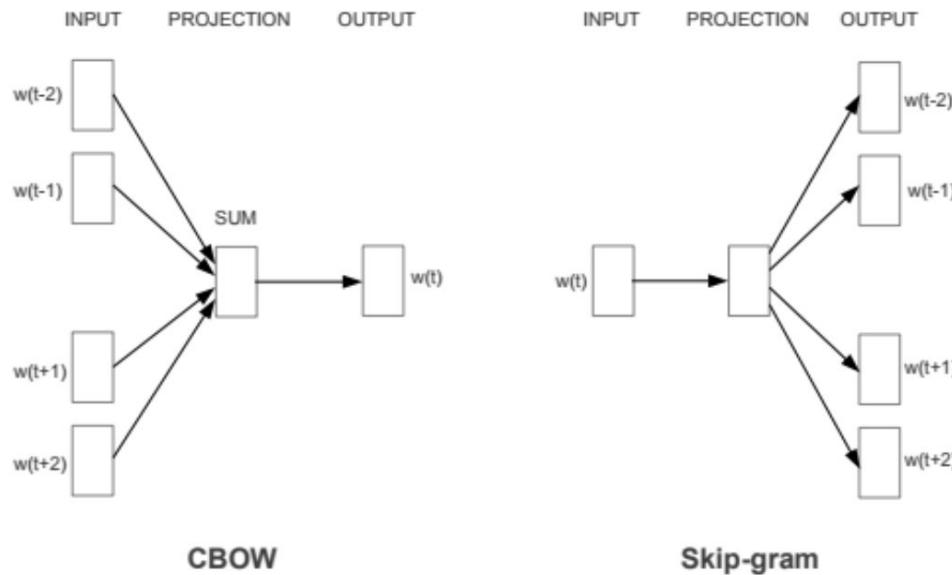
Word2vec

Word2vec é uma técnica recente desenvolvida para encontrar representações densas significativas de palavras. Ele se baseia na aplicação de redes neurais para **prever** palavras em um texto a partir de um determinado **contexto**

Existem dois principais modelos, o **CBOW** e o **Skip Gram**

Estado da arte da representação vetorial

Word2vec



A idéia no CBOW é prever uma palavra através das palavras no contexto (janela de n paavras)

Já no Skip-gram, a idéia se inverte. Tenta-se estimar o contexto através de uma palavra

Estado da arte da representação vetorial

Word2vec

Vantagens do Word2vec

Palavras semanticamente semelhantes aparecem próximas no reticulado;

Múltiplos graus de similaridade (Palavras com terminações semelhantes, sentidos semelhantes, mesmo radical, etc...);

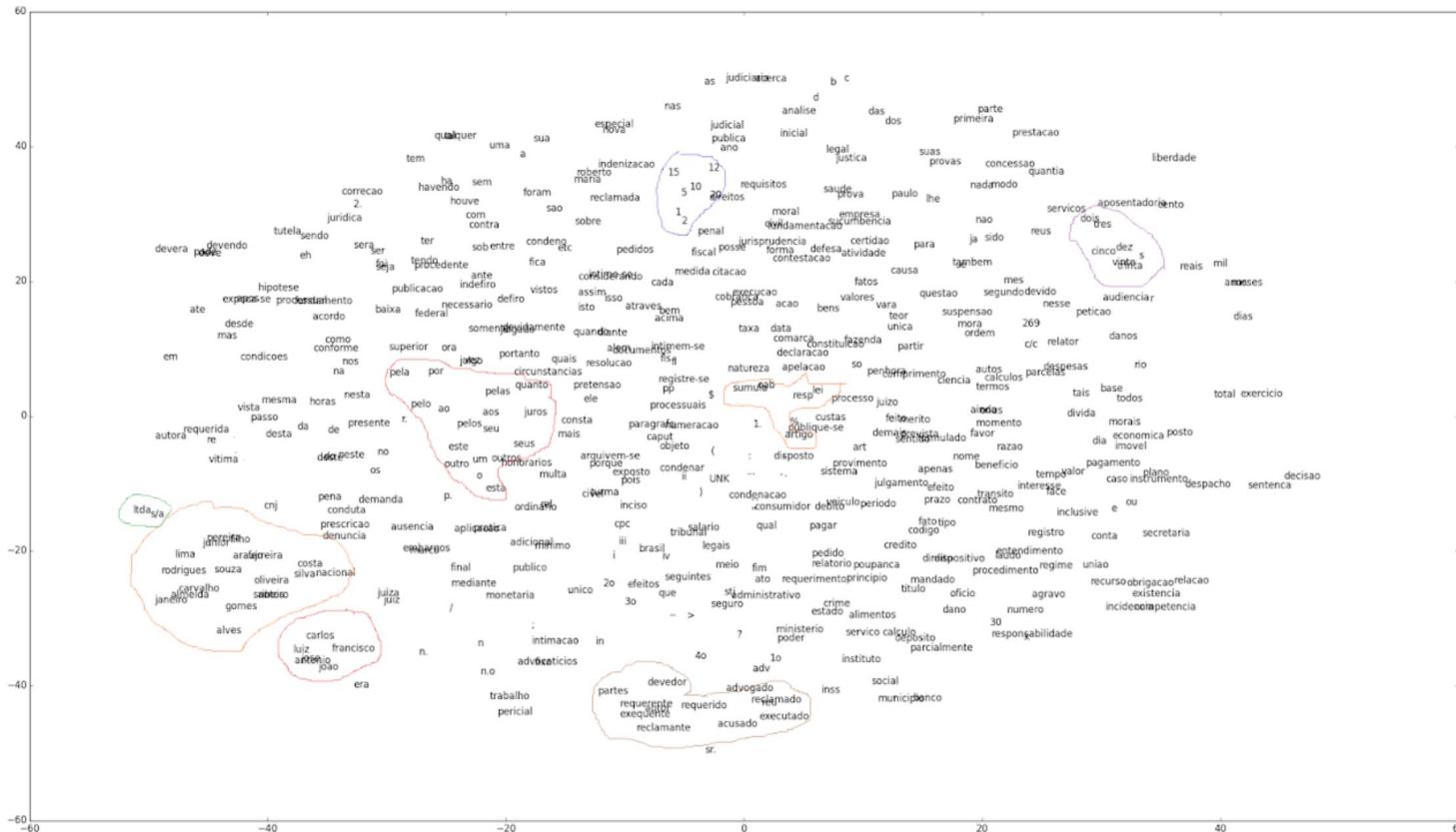
Operações com os vetores resultantes fazem sentido;

Rei - Homem + Mulher = Rainha

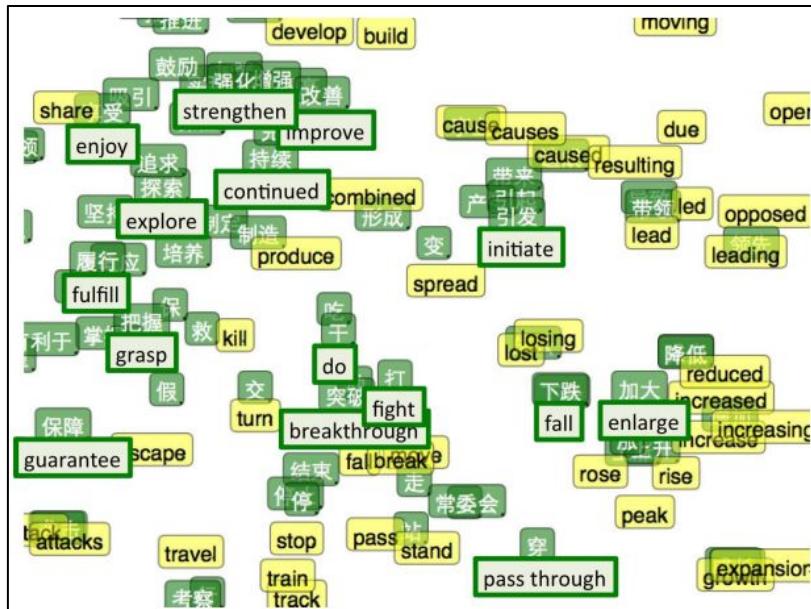
Alemanha - Berlim + França = Paris

São treinadas sobre dados não rotulados;

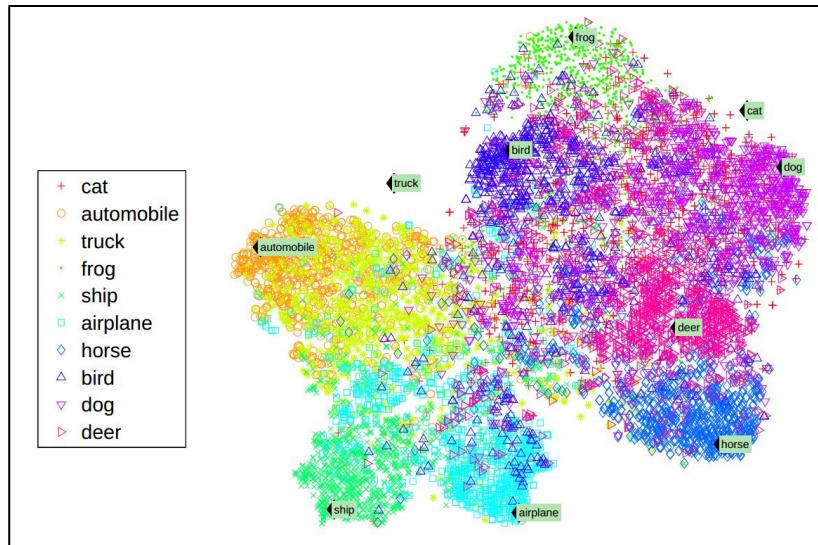
Exemplo de resultado com Texto Jurídico



Aplicações de Word2vec



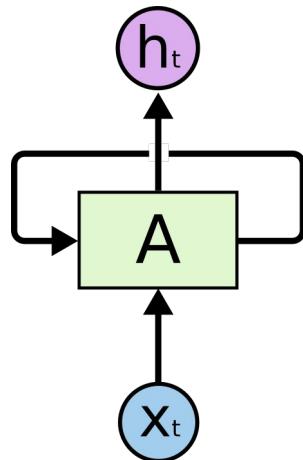
Tradução de máquina



Zero shot learning

Estado da arte em classificação de texto

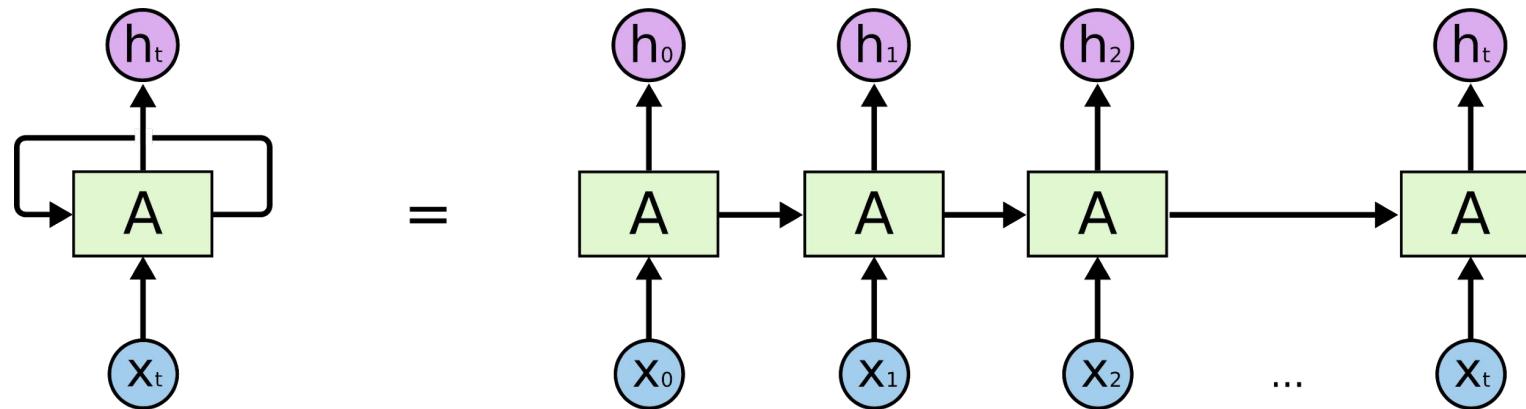
As melhores técnicas de processamento de texto da atualidade se baseiam em **redes recorrentes**



Fonte: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

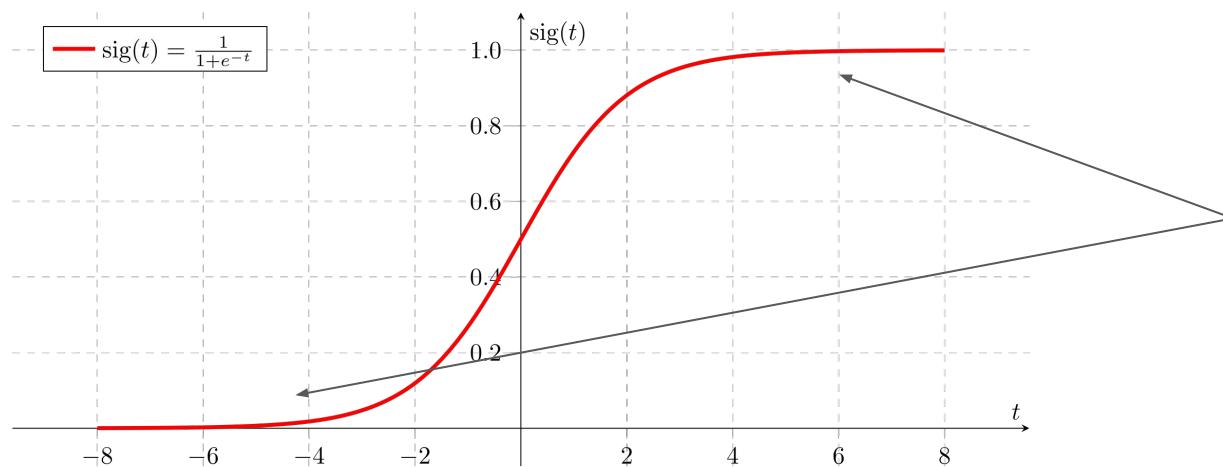
Estado da arte em classificação de texto

Redes recorrentes podem ser vistas como a repetição de uma mesma rede em sequência



Problemas de redes recorrentes

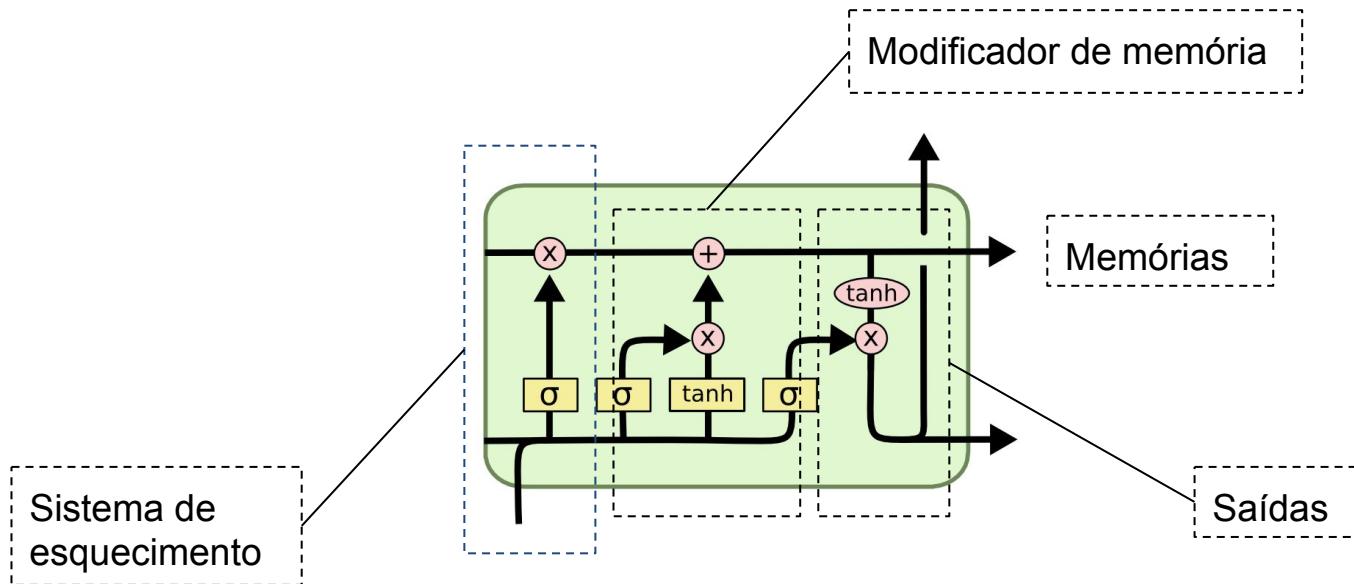
O principal problema de redes recorrentes é o **esquecimento catastrófico**



Em pontos extremos, o valor da função não se altera significativamente

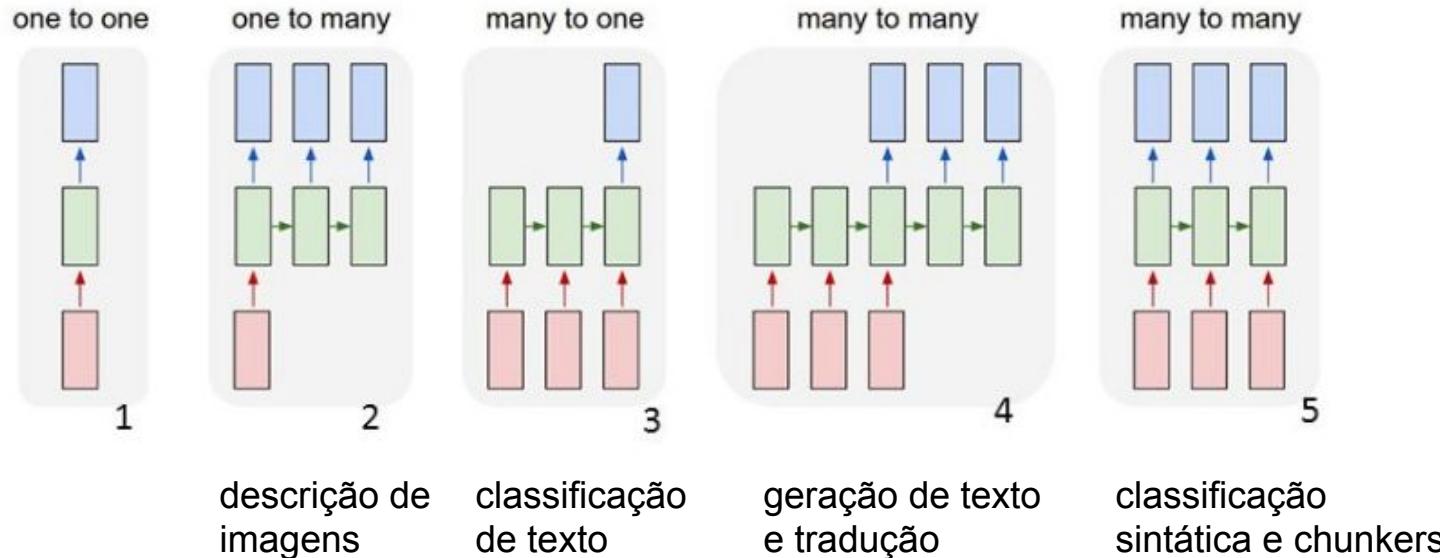
LSTM

Redes LSTM surgiram como uma proposta para mitigar este problema



LSTM

Redes LSTM podem funcionar em várias configurações e resolver diversos problemas de NLP

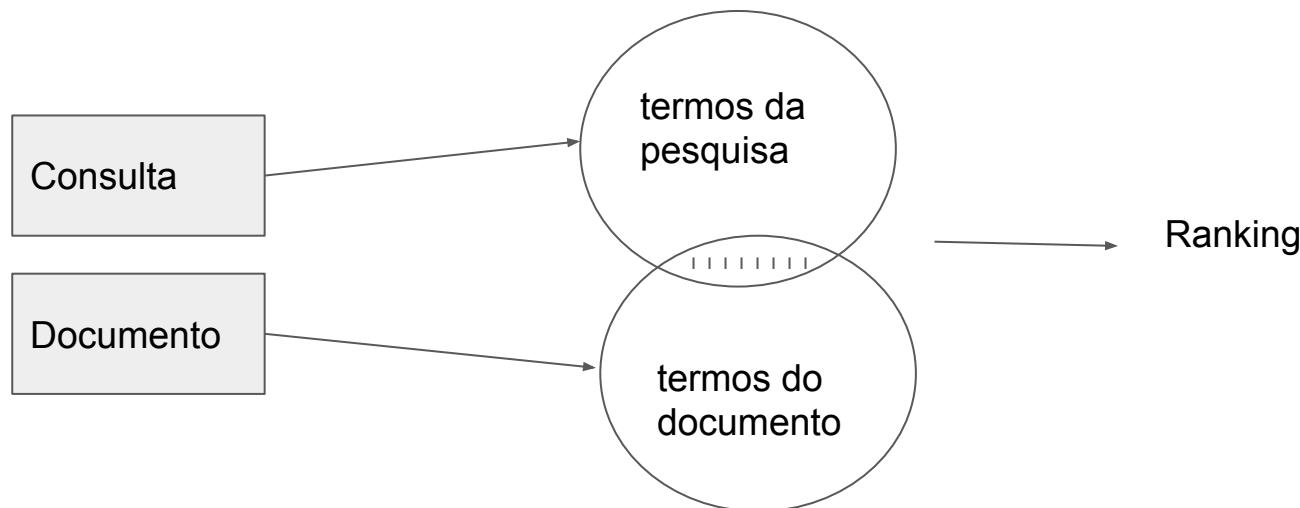


Recuperação inteligente de informação

As representações que discutimos até aqui podem ser utilizadas para fazer **recuperação inteligente** de documentos, por exemplo, em buscadores.

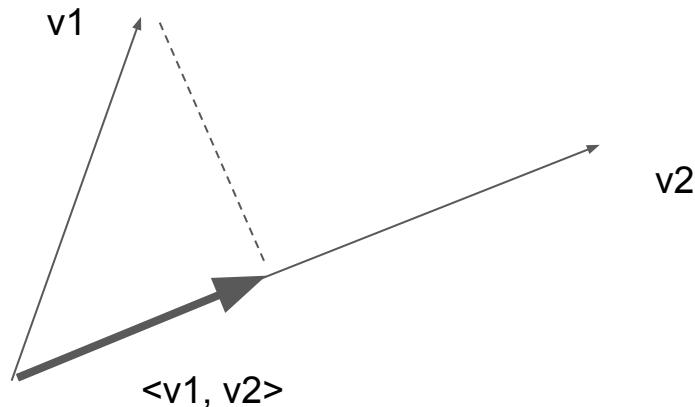
Busca inteligente

De maneira geral, um sistema de busca precisa computar uma medida de similaridade entre dois documentos. Uma forma de fazer isto é verificar a intersecção entre os termos da pesquisa e do documento sendo analisado



Busca inteligente

Caso seja possível representar os documentos e as pesquisas como vetores numéricos, podemos realizar uma medida de similaridade através do produto interno entre os dois vetores



Problemáticas na criação de projetos em NLP

Para a **língua inglesa**, já dispomos de muitos classificadores treinados com um desempenho muito bom. Ex: POS taggers e classificadores de nome entidade.

No entanto, para português, isso não acontece, de forma que o cientista de dados terá que treinar classificadores básicos do zero.

Problemáticas na criação de projetos em NLP

Ambiguidade na interpretação:

Eu vi um pássaro na árvore com um binóculo

Interpretação 1: Existe um pássaro na árvore, e eu estou observando o mesmo com um binóculo

Interpretação 2: Existe um pássaro na árvore, eu estou observando o mesmo e ele tem um binóculo

Interpretação 3: Existe um pássaro , ele está em uma árvore e o binóculo também está na árvore

Interpretação 4: Eu estou na árvore e eu vi um pássaro usando um binóculo

Para uma máquina, todas essas interpretações fazem sentido

Problemáticas na criação de projetos em NLP

Por motivos como esses, é bem mais fácil extrair pedaços de um texto com determinado significado do que propriamente extrair significado de um trecho qualquer de informação.

Fonte de recursos para NLP em Português

Na USP, existe o grupo de pesquisa **NILC** que disponibiliza diversos recursos úteis para o desenvolvimento de aplicações NLP.

<http://www.nilc.icmc.usp.br/nilc/tools/nilctaggers.html>

<http://www.nilc.icmc.usp.br/nilc/index.php>

<http://www.nilc.icmc.usp.br/nilc/index.php/tools-and-resources>

Problemáticas na criação de projetos em NLP

Outro problema que surge para projetos NLP é a ausência de textos **classificados** em português para o contexto que desejamos trabalhar. Desta forma, torna-se necessário que a equipe desenvolva uma ferramenta para criar este tipo de base de dados

Pipeline típico de um projeto em NLP

Fase 1: Em geral, torna-se necessário o desenvolvimento de uma aplicação onde os primeiros usuários do sistema (ou desenvolvedores) possam fazer marcações no texto ou classificações manuais. Esta aplicação será responsável por alimentar os primeiros classificadores a serem desenvolvidos e também permitirá a indicação de erros cometidos pelas primeiras versões do mesmo.

Pipeline típico de um projeto em NLP

Exemplo de marcação

Relatório

Segue o relatório, maria alega que joão roubou 5
reais de sua carteira.....

... Pede gratuidade da justiça e restituição total ...

Eu, o juiz fulano de tal, declaro joão como
culpado

Nomes de
pessoas

pretensão
do autor

Dispositivo

Pipeline típico de um projeto em NLP

Observe que podem haver marcações dentro de outras. Dessa forma, é natural representar um texto classificado através de uma estrutura XML. No entanto, quando estivermos realizando a classificação de um texto novo, algumas **inconsistências** podem ocorrer e precisam ser tratadas.

Pipeline típico de um projeto em NLP

Casos extremos



As tags precisarão ser ajustadas para gerar um **XML válido**

Pipeline típico de um projeto em NLP

Outras ferramentas necessárias

Cada classificador desenvolvido pode retornar o seu resultado de uma forma diferente. Dessa forma, para montar o XML final, se faz necessário desenvolver um algoritmo para encontrar pedaços de texto em um texto maior de maneira eficiente e robusta.

Ex: Suponha que para desenvolver o nosso identificador de leis, removemos todos os caracteres estranhos do texto. Porém, no texto original temos escrito lei maria-da-penha.

Pipeline típico de um projeto em NLP

Fase 2: Aquisição inicial da base de dados

Nesta etapa, pode ser realizada a classificação manual de alguns textos com a ferramenta desenvolvida, ou uma classificação simples baseada em **expressões regulares**. Por exemplo, caso se deseje fazer um classificador de resultado da sentença jurídica, pode-se fazer um dicionário de expressões que indicam um determinado resultado

Pipeline típico de um projeto em NLP

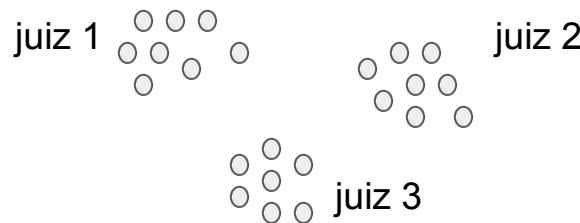
Fase 2: Aquisição inicial da base de dados

Também é possível obter conjuntos de treinamento a partir de mineração de páginas online. Por exemplo, podemos construir bases de dados de nomes de pessoas e nomes de empresas e construir um classificador de pessoa física x pessoa jurídica.

Pipeline típico de um projeto em NLP

Fase 2: Aquisição inicial da base de dados

Outra opção é representar os documentos através de vetores numéricos (de acordo com as técnicas que mostraremos) e tentar identificar **grupos** através de aprendizado não supervisionado. Estes grupos podem indicar classes ou pelo menos indicar onde devemos pegar amostras para realizar a classificação manual



Dessa forma, podemos montar uma sub-amostra **estatisticamente** relevante

Pipeline típico de um projeto em NLP

Fase 3: Desenvolvimento dos classificadores iniciais

Nesta etapa, podemos utilizar as primeiras bases construídas para desenvolver protótipos de classificadores

Pipeline típico de um projeto em NLP

Fase 4: Testes dos classificadores e re-treinamento

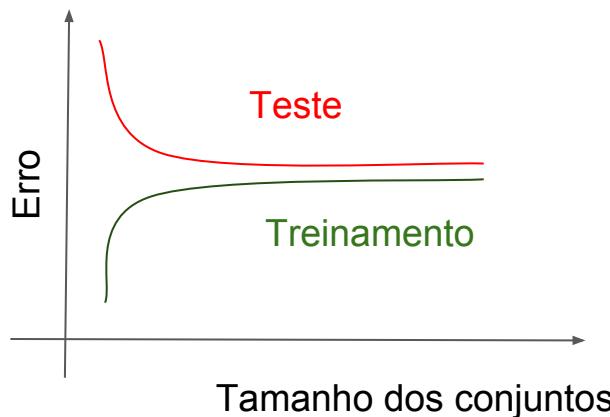
Nesta etapa, aplicam-se os classificadores desenvolvidos na fase 3 e, juntamente com a aplicação desenvolvida, os erros serão corrigidos e os classificadores, re-treinados.

Pipeline típico de um projeto em NLP

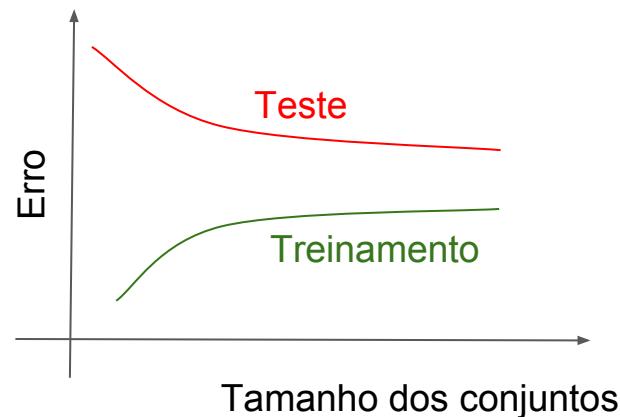
Fase 5: Análise de overfitting x underfitting

Neste momento, uma vez que os classificadores estão sendo aplicados e a base de dados está **aumentando**, podemos traçar as curvas de aprendizado e efetivamente verificar se a **complexidade** dos modelos deve ser aumentada

Underfitting



Overfitting



Classificação de textos

Suponha que você trabalhe no IMDB e deseja **classificar** uma review de filme em **positiva** ou **negativa**

User Reviews

★★★★★ Amazing!

16 December 2006 | by jellienellie818 (United States) – See all my reviews

I have been a fan of Will Smith for years and I have to say this may be his best film yet! "The Pursuit of Happiness" is just a wonderful (based on a true) story, full of adventure, hope, and pain. I saw the movie last night in a packed theater. Big Willie Weekend has returned, and for good reason! It's a great movie to see during the holidays and definitely a tear-jerker! Perfect for a date, a night out with friends, or even with family. If you ever thought Will Smith really couldn't act (and shame on you!), you'll think otherwise once you see this movie. You can really feel what he's going through just by looking in his eyes. And Jaden Smith is too adorable! Their on screen chemistry is almost unbearable to watch! Go see this movie! Great acting, great directing, great writing...you won't regret it!

156 of 192 people found this review helpful. Was this review helpful to you?

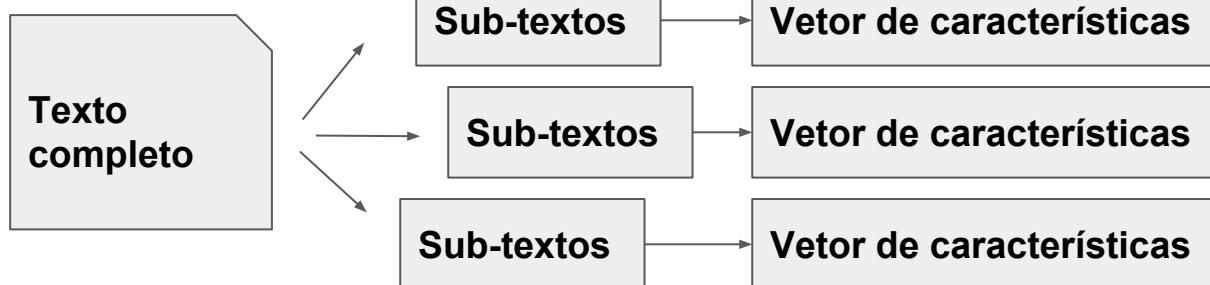
[Review this title](#) | [See all 574 user reviews »](#)

Classificação de textos

A classificação de texto é uma das principais técnicas que estudaremos aqui. Ela pode ser utilizada para classificar um **texto inteiro** em classes (ex: procedente, improcedente, etc....) ou até mesmo identificar partes relevantes de um texto (ex: dispositivo, fato, etc...)

Classificação de textos

Classificação de um texto inteiro

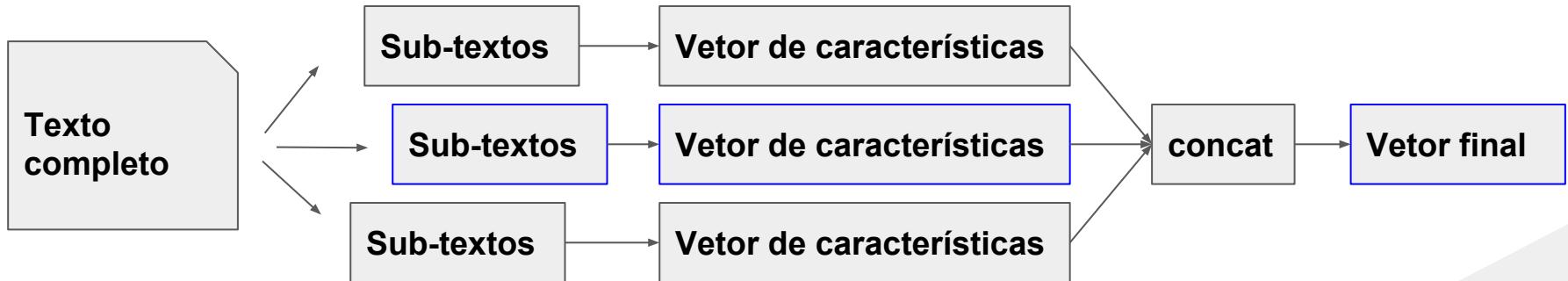


Geralmente a divisão em sub-textos ocorre por segmentação em sentenças

Classificação de textos

Classificação de partes de um texto

A classificação de uma única parte do texto pode usar o vetor de features das partes adjacentes



Classificação de textos

A primeira coisa que deve ser feita é transformar o texto em uma **matriz numérica** para ser empregada nos classificadores que conhecemos. Além disso, seria interessante que a matriz tivesse **tamanho fixo**

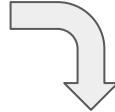
User Reviews

★★★★★ **Amazing!**
16 December 2006 | by jellienellie818 (United States) – See all my reviews

I have been a fan of Will Smith for years and I have to say this may be his best film yet! "The Pursuit of Happiness" is just a wonderful (based on a true) story, full of adventure, hope, and pain. I saw the movie last night in a packed theater. Big Willie Weekend has returned, and for good reason! It's a great movie to see during the holidays and definitely a tear-jerker! Perfect for a date, a night out with friends, or even with family. If you ever thought Will Smith really couldn't act (and shame on you!), you'll think otherwise once you see this movie. You can really feel what he's going through just by looking in his eyes. And Jaden Smith is too adorable! Their on screen chemistry is almost unbearable to watch! Go see this movie! Great acting, great directing, great writing...you won't regret it!

156 of 192 people found this review helpful. Was this review helpful to you?

[Review this title](#) | See all 574 user reviews >



[0 1 32 50 35 10]

Classificação de textos

Bag of words

Uma das técnicas mais simples e efetivas. Cada feature representa quantas vezes a palavra em questão aparece

texto 1: ola tudo bem

texto 2: tudo certo meu senhor

texto 3: o senhor está bem?

Classificação de textos

Bag of words

Uma das técnicas mais simples e efetivas. Cada feature representa quantas vezes a palavra em questão aparece

texto 1: ola tudo bem

texto 2: tudo certo meu senhor

texto 3: o senhor está bem?

vocabulário: {ola, tudo, bem, certo, meu, senhor, está, o}

Classificação de textos

Bag of words

Uma das técnicas mais simples e efetivas. Cada feature representa quantas vezes a palavra em questão aparece

texto 1: ola tudo bem

ola	tudo	bem	certo	meu	senhor	está	o
1	1	1	0	0	0	0	0

Classificação de textos

Bag of words

Uma das técnicas mais simples e efetivas. Cada feature representa quantas vezes a palavra em questão aparece

texto 2: tudo certo meu senhor

ola	tudo	bem	certo	meu	senhor	está	o
0	1	0	1	1	1	0	0

Classificação de textos

Bag of words

Uma das técnicas mais simples e efetivas. Cada feature representa quantas vezes a palavra em questão aparece

texto 3: o senhor está bem?

ola	tudo	bem	certo	meu	senhor	está	o
0	0	1	0	0	1	1	1

Classificação de textos

Bag of words

Uma das técnicas mais simples e efetivas. Cada feature representa quantas vezes a palavra em questão aparece

texto 3: o senhor está bem bem?

ola	tudo	bem	certo	meu	senhor	está	o
0	0	2	0	0	1	1	0

Classificação de textos

Melhorias na representação Bag of words

- Usar um **stemmer** ou **lemmatizer**:

Transforma todas as palavras na sua forma radical

Exemplo:

cozinhar, cozinhando e cozinharemos serão todos considerados uma palavra só

stemmer: cozinh

lemmatizer: cozinar

Dessa forma, reduzimos a dimensionalidade das nossas matrizes

Classificação de textos

Melhorias na representação Bag of words

- Remover stopwords:

Stopwords são palavras muito comuns que não possuem grande significado

Exemplo:

a, o, pelo, como, e, à, de,.....

Pode-se utilizar **listas já construídas** de stopwords

Classificação de textos

Representações alternativas:

1. Em vez de utilizarmos o número de vezes que uma palavra apareceu, podemos utilizar a proporção de cada palavra;
2. Podemos utilizar também features binárias, indicando se uma palavra apareceu ou não.

Classificação de textos

Melhorias na representação Bag of words

Identificar palavras mais relevantes através da
métrica **Ganho de informação**

Classificação de textos

Bag of words

De posse das matrizes bag of words construídas, podemos alimentar os classificadores que estudamos, mas será que existe **uma forma melhor de contabilizar as palavras?**

Classificação de textos

Tf-idf

É uma forma de tentar identificar de forma automática quais palavras são relevantes e quais são stopwords

Cada feature, em vez de identificar quantas vezes a palavra aparece, representará a seguinte grandeza

$$feat_i = tf/idf$$

Classificação de textos

Tf-idf

$$feat_i = tf / idf$$

O termo **tf** é chamado de *term frequency* e é a contagem de quantas vezes a palavra aparece no texto, similar ao que ocorre com a técnica Bag of Words

Já o termo **idf** é chamado de inverse *document frequency* e é dado por

$$Idf = \log \left[\frac{(\text{Número total de documentos}) + 1}{(\text{Número de documentos com a palavra}) + 1} \right] + 1$$

Classificação de textos

Tf-idf

$$feat_i = tf/idf$$

$$Idf = \log\left[\frac{(Número\ total\ de\ documentos) + 1}{(Número\ de\ documentos\ com\ a\ palavra) + 1}\right] + 1$$

Termos muito comuns em todos os textos (stopwords) terão o termo idf bem alto, logo o valor da feature associada àquela palavra irá cair

A utilização da técnica Tf-idf torna a diferença entre palavras relevantes e stopwords **mais suave**, tendendo a **melhorar os resultados**

Classificação de textos

Tf-idf

$$feat_i = tf/idf$$

$$Idf = \log\left[\frac{(Número\ total\ de\ documentos) + 1}{(Número\ de\ documentos\ com\ a\ palavra) + 1}\right] + 1$$

Termos muito comuns em todos os textos (stopwords) terão o termo idf bem alto, logo o valor da feature associada àquela palavra irá cair

A utilização da técnica Tf-idf torna a diferença entre palavras relevantes e stopwords **mais suave**, tendendo a **melhorar os resultados**

Análise de semântica latente

Além das técnicas mencionadas até agora, existe uma metodologia chamada **análise de semântica latente** que permite a construção de vetores menores para a classificação de textos e ainda permite construir **representações densas de palavras**

Análise de semântica latente

Além das técnicas mencionadas até agora, existe uma metodologia chamada **análise de semântica latente** que permite a construção de vetores menores para a classificação de textos e ainda permite construir **representações densas de palavras**

A representação bag of words esbarra em dois grandes problemas técnicos

Problemas do bag of words

Problema 1: Polissemia

Um problema de polissemia ocorre quando uma única palavra têm múltiplos significados. Ex: gato (o animal ou uma ligação ilegal de energia)

Problema 2: Sinonímia

Um problema de sinonímia ocorre quando duas palavras diferentes podem ter o mesmo significado (cachorro e canino)

Problemas do bag of words

Note que a sinonímia e a polissemia dependem de contexto.

Uma forma de lidar com a polissemia é através da **análise sintática** do texto. Falaremos disto quando discutirmos POS taggers. Para resolver a sinonímia, podemos realizar **combinações lineares** de palavras

$$\text{palavra_abstrata} = 0.5*\text{cachorro} + 0.2*\text{canino} + 0.4*\text{lobo} + \dots$$

A criação dessas palavras abstratas é chamada de análise de semântica latente.

Matemática da análise de semântica latente

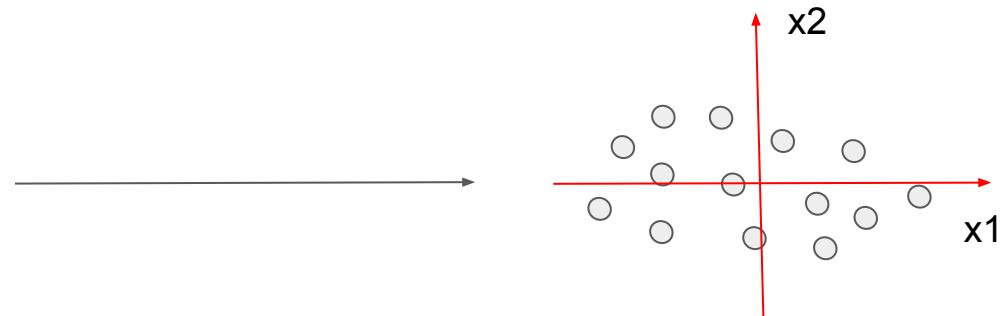
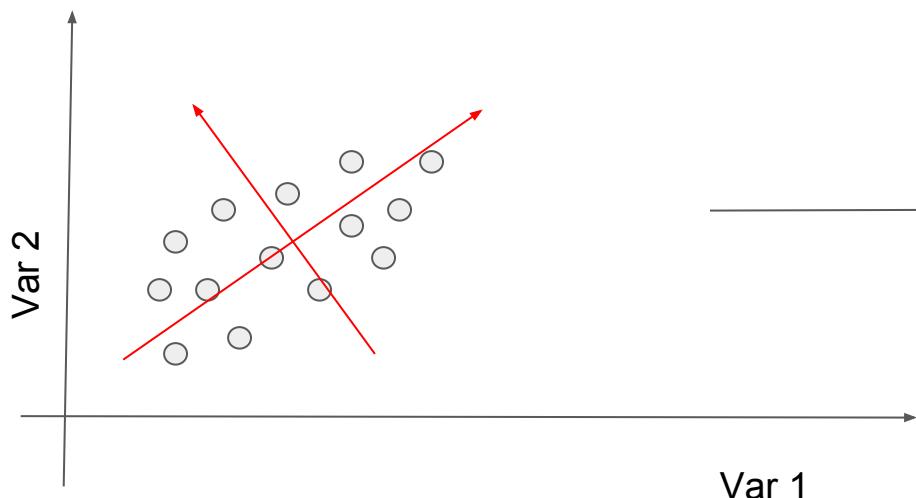
A técnica matemática utilizada em análise de semântica latente é conhecida como **SVD**, que é bastante similar ao **PCA**, discutido anteriormente. O primeiro passo neste processo é a criação de uma matriz de termo x documento.

Matriz termo x documento

Basicamente, se temos na nossa base de dados um conjunto de **N** documentos que contém um vocabulário de **M** palavras, devemos construir uma matriz **N x M**, onde cada linha representa um documento e cada elemento na matriz representa quantas vezes a palavra correspondente apareceu no documento em questão.

SVD

O procedimento SVD vai, de maneira similar ao PCA, criar um novo sistema de coordenadas onde cada variável é independente da outra

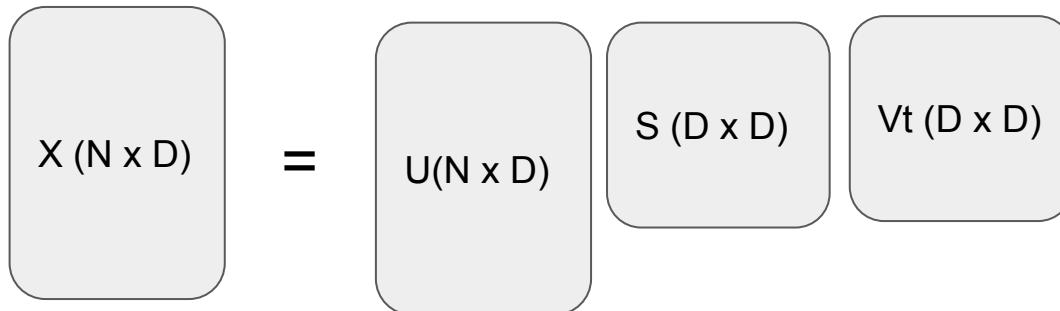


SVD

O **SVD** trata de um método em álgebra linear para representar uma matriz X de dimensão ($N \times M$) por um produto de três matrizes:

$$X = U.S.V^T$$

Podemos visualizar este produto graficamente

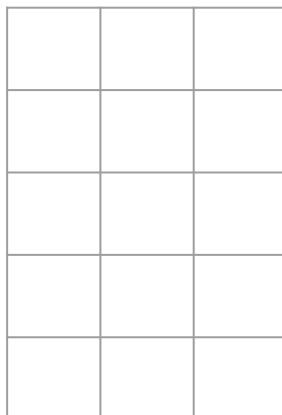


SVD

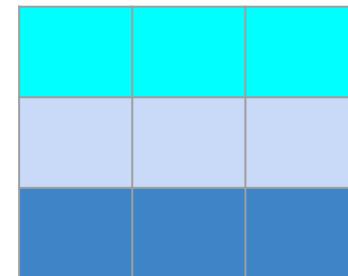
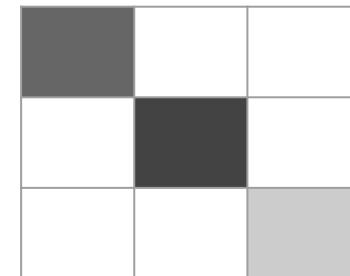
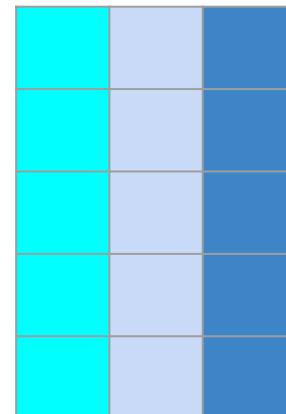
O **SVD** trata de um método em álgebra linear para representar uma matriz X de dimensão ($N \times M$) por um produto de três matrizes:

$$X = U.S.V^T$$

Podemos visualizar este produto graficamente



=



SVD

O **SVD** trata de um método em álgebra linear para representar uma matriz X de dimensão (NxM) por um produto de três matrizes:

$$X = U.S.V^T$$

A matriz S é uma matriz diagonal e indica o desvio padrão de cada nova variável. Esses valores também são chamados de auto-valores das novas variáveis

SVD

O **SVD** trata de um método em álgebra linear para representar uma matriz X de dimensão (NxM) por um produto de três matrizes:

$$X = U.S.V^T$$

A matriz V^T nada mais são do que os auto-vetores de cada um dos valores de S, ou seja, são as **representações densas de palavras**

Cyan	Cyan	Cyan
Light Blue	Light Blue	Light Blue
Dark Blue	Dark Blue	Dark Blue

SVD

Se desejamos transformar um vetor x no espaço inicial em um vetor z no novo espaço, basta realizar a seguinte operação

$$z = V^T x$$

Desta forma, fica simples realizar a conversão entre os espaços de coordenadas. Estas matrizes transformadas podem ser utilizadas para realizar **comparações entre documentos** em um espaço reduzido.

Por que SVD faz sentido como feature?

Imagine que desejamos fazer um sistema de recomendação de filmes, e séries, estilo Netflix

Uma abordagem é recomendar ao cliente filmes que são parecidos com os filmes que o usuário já vem assistindo

Por que SVD faz sentido como feature?

Suponha que cada cliente classifique cada filme dando estrelas para o mesmo, e suponha que esta nova vá de uma até 5 estrelas.

Podemos fazer uma espécie de matriz documento x termo colocando quantas estrelas cada usuário deu para cada filme

Filme	João	Felipe	Carlos
Matrix	3	4	1
Pokemón	5	2	2
Game of Thrones	5	4	3

Por que SVD faz sentido como feature?

Através do SVD, serão criadas várias variáveis abstratas que podem fazer muito mais sentido do que as variáveis anteriores

Filme	Filme de tecnologia	Fantasia	Tem o ator x?
Matrix	5.67	3.4	0
Pokemón	0.7	8.9	0.0
Game of Thrones	0.1	5.6	3.0

Pos tagging

O pos tagging é uma etapa necessária em muitas pipelines de processamento de linguagem natural. A ideia é classificar sintaticamente cada palavra (token) de uma frase

Ele faz parte de uma classe de problemas que envolve classificar cada palavra de um texto em determinado conjunto de classes (no caso, as diversas funções sintáticas possíveis)

Pos tagging

O pos tagging é uma etapa necessária em muitas pipelines de processamento de linguagem natural. A ideia é classificar sintaticamente cada palavra (token) de uma frase

Ex:

Olá, meu nome é João da Silva e trabalho no Banco

(Olá, I) (meu, PPS) (nome, N) (é, VLIG) (João, NP) (da, NP) (Silva, NP)
(e, CONJCOORD) (trabalho, N) (na, PREP+ART) (Banco, N)

Pos tagging

Em geral, o pos tagger é um classificador Bayesiano, cujas features são a palavra a ser classificada e as palavras nos arredores

Unigram tagger: Usa apenas a palavra atual $P[\text{tag} = t \mid \text{palavra} = x]$

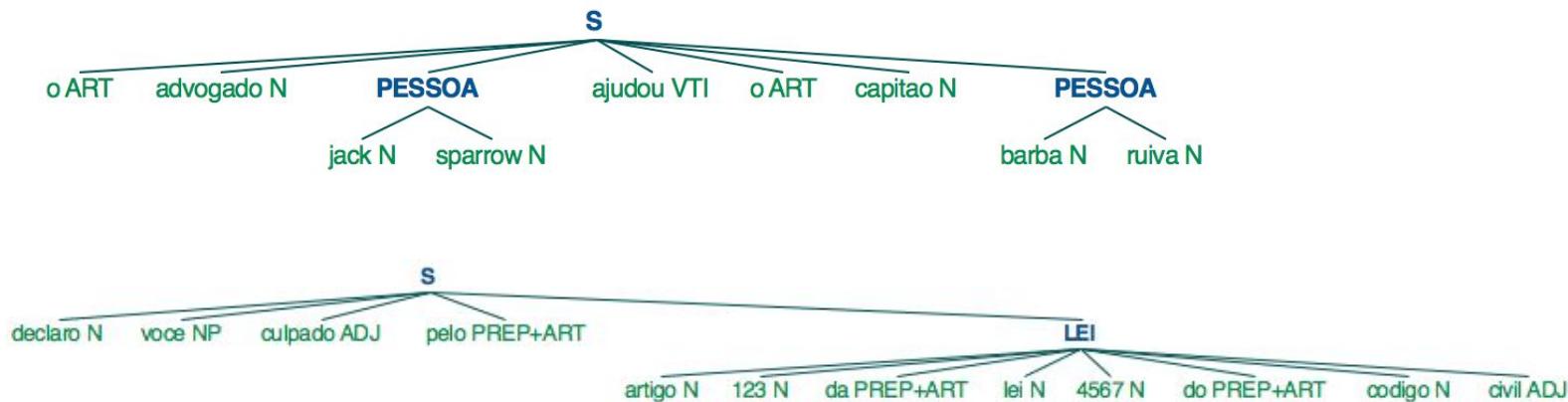
Bigram tagger: Usa a palavra atual e a palavra anterior

$P[\text{tag} = t \mid \text{palavra} = x \text{ e } \text{palavra_anterior} = y]$

e assim, podemos construir diferentes pos taggers de acordo com o número de palavras que queremos considerar

Chunkers

Chunkers são capazes de encontrar entidades com significado em um texto, exemplo:



Chunkers

Notação IOB

Basicamente, chunkers classificam as palavras (tokens) em três classes:

O - fora de um chunk

B - início de um chunk

I - interior de um chunk

Chunkers

Notação IOB

O - fora de um chunk

B - início de um chunk

I - interior de um chunk

Ex: meu nome é joão da silva e trabalho no Banco

(meu, O) (nome, O) (é, O) (joão, B) (da, I) (silva, I) (e, O) (trabalho, O)

(na, O) (Banco, O)

Chunkers

Funcionamento de um chunker

Muitas vezes um chunker usa uma arquitetura **sequencial** similar a uma rede recorrente, ou seja, utilizamos a classificação anterior como feature. A diferença é que em tempo de treinamento, simplificamos a análise e sempre inserimos a predição correta para o caso anterior como feature para o caso atual.

Chunkers

Features

O classificador roda em cada palavra classificando em I, O ou B. Aqui temos algumas possíveis features

prefix3: primeiras três letras da palavra

suffix3: últimas três letras da palavra

pos: classificação sintática da palavra

word: a palavra em si

prevtag: a classificação IOB da palavra anterior

prevword: a palavra anterior

prevpos: a classificação sintática da palavra anterior

nextword: a próxima palavra

nextpos: a classificação sintática da próxima palavra

Chunkers

Criação da matriz numérica

Para converter as features anteriores em uma matriz numérica, é criado um vetorizador Bag of Words para cada uma das features

O vetor de features resultante pode ficar com mais de 30k elementos.
Assim, o Chunker é um classificador relativamente lento.