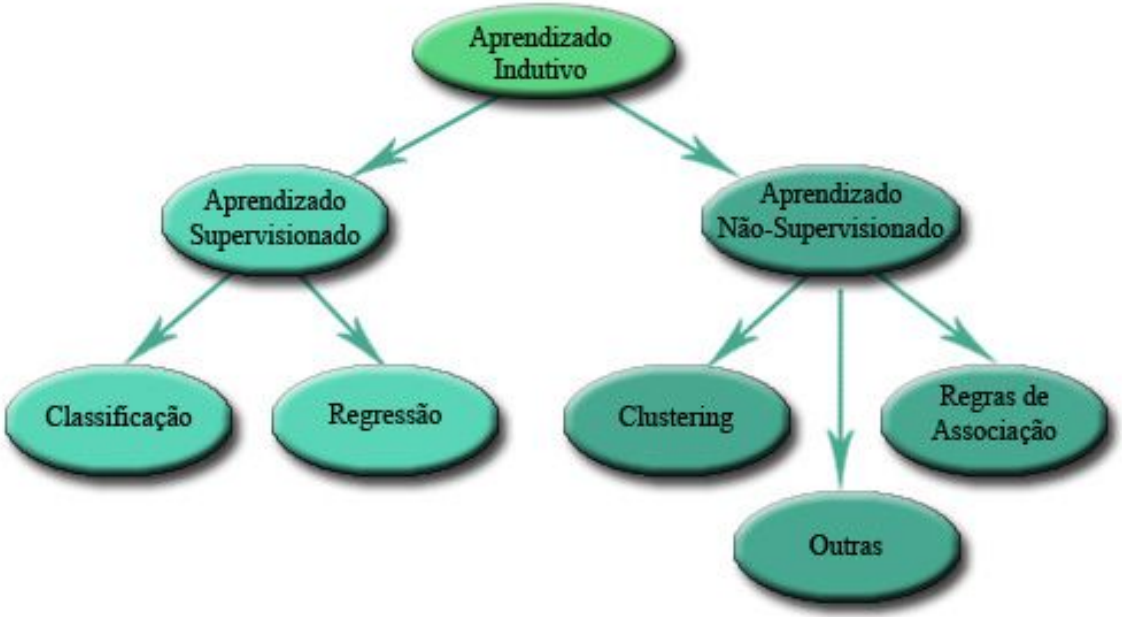
Aprendizado de Máquina

Tarefas de Aprendizado

- Preditivas - a meta é encontrar uma função (modelo ou hipótese) a partir dos dados de treinamento que possa ser utilizada para prever um rótulo ou atributo alvo. (aprendizado supervisionado)
- Descritivas - a meta é explorar e descrever um conjunto de dados. Algoritmos de AM utilizados não fazem uso de atributos de saída (aprendizado não supervisionado)

Hierarquia de Aprendizado



Aprendizado de Máquina

Classificadores

K vizinhos mais próximos

K nearest neighboors – KNN

K vizinhos mais próximos

K nearest neighboors – KNN

Classificar consiste em determinar a que classe um objeto pertence, dados os valores de um conjunto de atributos do objeto.
Matematicamente, classificar é aplicar uma função que dados os valores dos atributos como entrada produz como resposta uma saída discreta, neste caso, cada objeto pertence a apenas uma classe entre um conjunto fixo de possibilidades

- filtragem de spam
- detectar se uma operação com cartão de crédito é fraudulenta ou não
- detectar se um conjunto de pixels é um rosto ou não

Aprendizado de Máquina

Classificadores

K vizinhos mais próximos

K nearest neighboors – KNN

K vizinhos mais próximos

K nearest neighboors – KNN

- O KNN foi proposto por Fukunaga e Narendra em 1975
- um dos classificadores mais simples de ser implementado
- de fácil compreensão e implementação
- pode obter bons resultados dependendo de sua aplicação

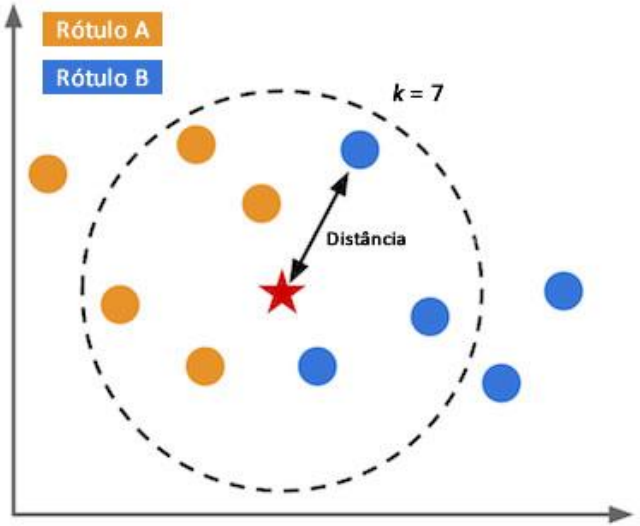
K vizinhos mais próximos

K nearest neighboors – KNN

O objetivo é determinar o rótulo de classificação de uma amostra baseado nas amostras vizinhas advindas de um conjunto de treinamento.

$$Q = (q_1, \cdots, q_n)$$

A FIgura 1 ilustra um problema de classificação com dois rótulos de classe e com k = 7. No exemplo, são aferidas as distâncias de uma nova amostra, representada por uma estrela, às demais amostras de treinamento, representadas pelas bolinhas azuis e laranjas. A variável k representa a quantidade de vizinhos mais próximos que serão utilizados para averiguar de qual classe a nova amostra pertence. Com isso, das sete amostras de treinamento mais próximas da nova amostra, 4 são do rótulo A e 3 do rótulo B. Portanto, como existem mais vizinhos do rótulo A, a nova amostra receberá o mesmo rótulo deles, ou seja, A.



Dois pontos chaves que devem ser determinados para aplicação do KNN são: a métrica de distância e o valor de k. Para métrica de distância a mais utilizada é a distância Euclidiana, descrita por:

$$D = \sqrt{(p_1 - q_1)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Em relação ao valor k, não existe um valor único para a constante, a mesma varia de acordo com a base de dados. É recomendável sempre utilizar valores ímpares/primos, mas o valor ótimo varia de base para base. Dependendo do seu problema você pode utilizar um algoritmo de otimização (PSO, GA, DE…) para encontrar o melhor valor para o seu problema.

K nearest neighboors

Resumidamente, a grande vantagem do KNN é sua abordagem simples de ser compreendida e implementada. Todavia, calcular distância é tarefa custosa e caso o problema possua grande número de amostras o algoritmo pode consumir muito tempo computacional. Além disso, o método é sensível à escolha do k. Veja um pseudocódigo do algoritmo:

```
1  inicialização:
2      Preparar conjunto de dados de entrada e saída
3      Informar o valor de k;
4  para cada nova amostra faça
5      Calcular distância para todas as amostras
6      Determinar o conjunto das k's distâncias mais próximas
7      O rótulo com mais representantes no conjunto dos k's
8      vizinhos será o escolhido
9  fim para
10 retornar: conjunto de rótulos de classificação
```

scikit-learn

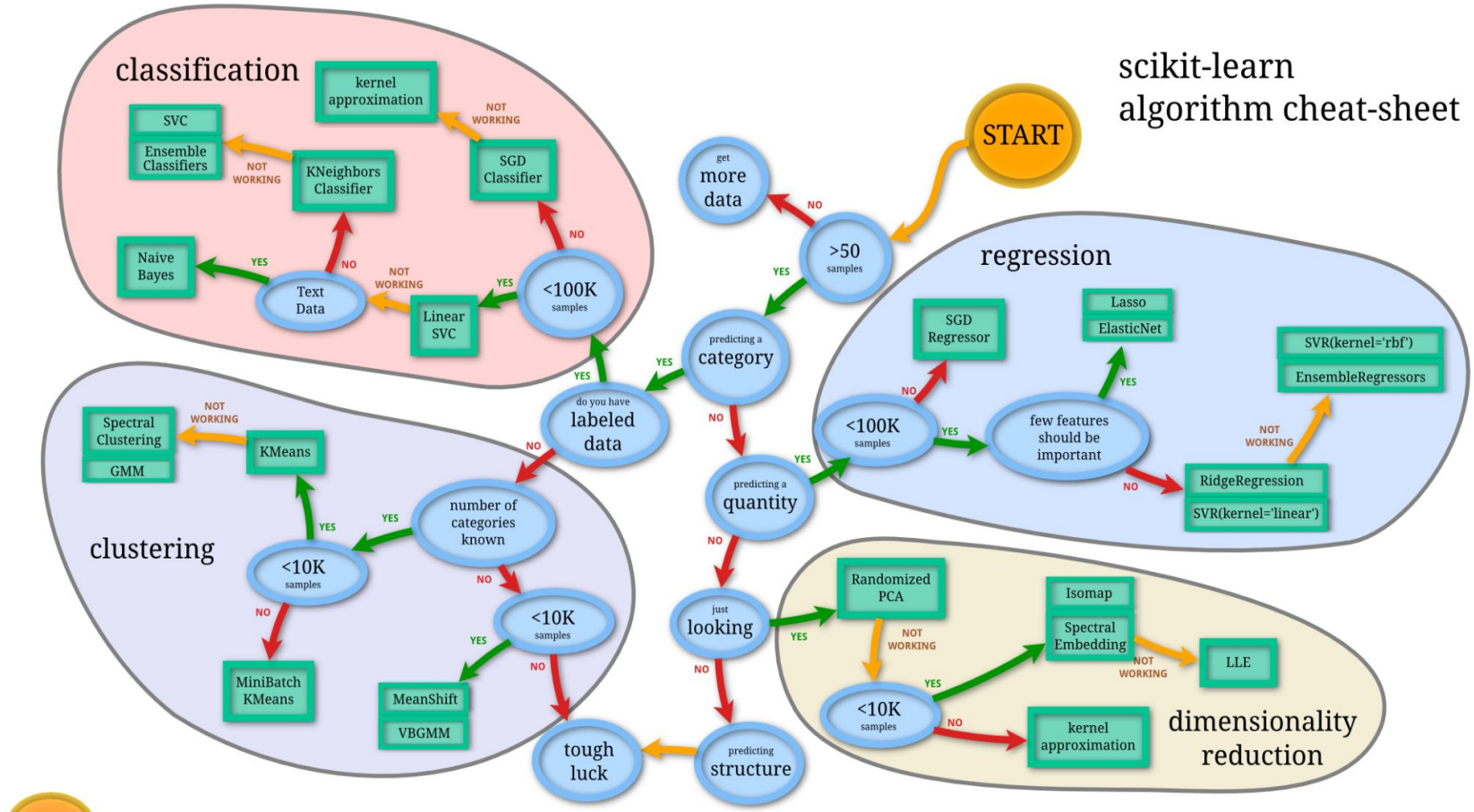
Machine Learning in Python

- Utilização fácil, intuitiva e replicável
- Documentação excelente
- Fácil integração com outros pacotes do Python como Pandas, Matplotlib, Numpy, etc
- Manutenção em dia
- Código aberto com licença BSD

Visão geral Scikit-Leaning

A biblioteca pode ser dividida em cinco módulos:

- Estimadores básicos (KNN, RNA, etc.) (fit e predict)
- Pre-processamento e transformadores (fit e transform)
- Avaliação de modelos (split, score)
- Pipelines
- Busca automática de parâmetros (GridSearchCV() e o RandomizedSearchCV())


scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
predicting a category
YES
NO
do you have labeled data
YES
NO
predicting a quantity
YES
NO
just looking
YES
NO
predicting structure
tough luck
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
YES
clustering
Spectral Clustering
GMM
NOT WORKING
KMeans
YES
number of categories known
YES
<10K samples
NO
MiniBatch KMeans
NO
<10K samples
YES
MeanShift
VBGMM
NO
regression
SGD Regressor
NO
<100K samples
YES
few features should be important
YES
Lasso
ElasticNet
NO
RidgeRegression
SVR(kernel='linear')
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
dimensionality reduction
Randomized PCA
YES
NOT WORKING
<10K samples
YES
Isomap
Spectral Embedding
NOT WORKING
LLE
NO
kernel approximation
Back
scikit learn

Aprendizado de Máquina

Regressão Linear

Regressão Linear Simples

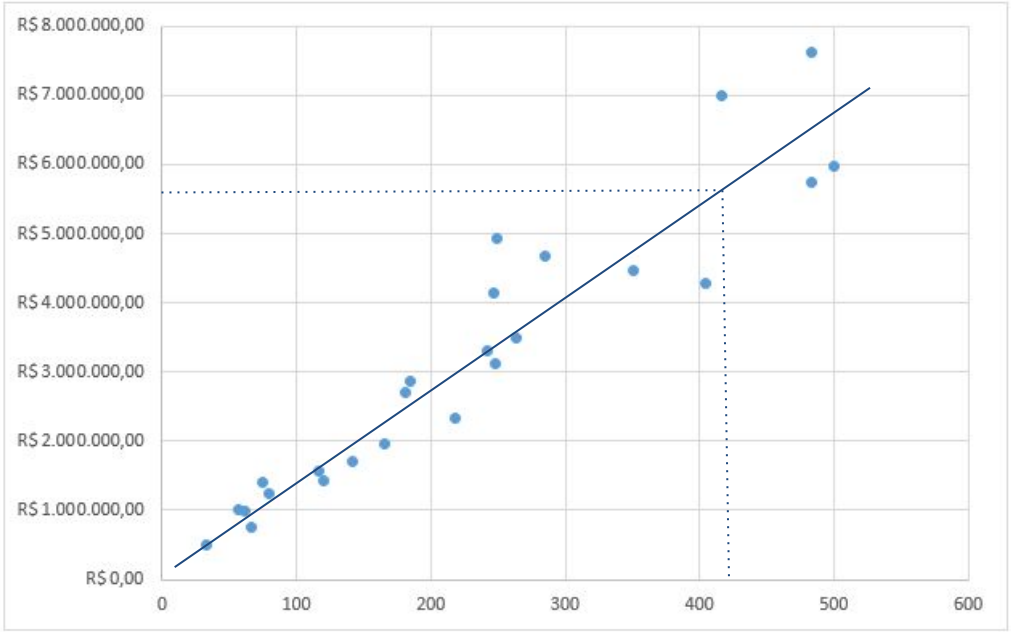
Uma outra tarefa importante de aprendizado é a regressão.
A diferença entre classificação e regressão é que nesta última a saída do sistema (**da função aprendida) é contínua**.
Por exemplo, prever o preço que um objeto terá quando vendido em um mercado, ou prever a pontuação média no ENEM dos alunos de uma escola, baseado na pontuação dos mesmos alunos em um simulado aplicado pela escola..

Regressão Linear Simples

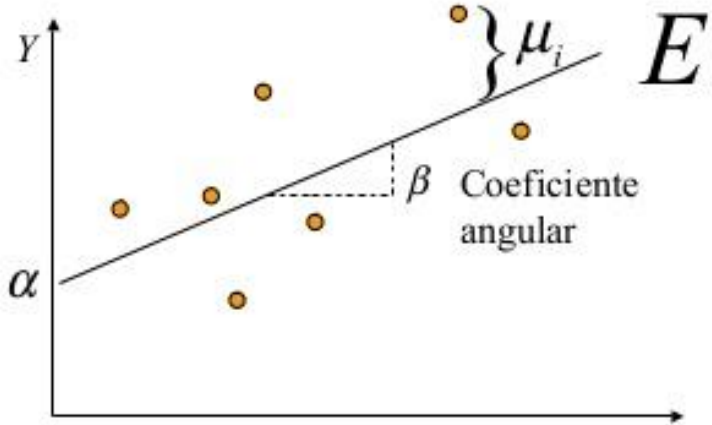
O objetivo da regressão:

- Avaliar uma possível dependência da variável y em relação a variável x.
- Expressar essa dependência por meio de uma equação da reta f(y)= ax + b

i	Tamanho(m2)	Valor da Casa
1	33	R$ 504.659,00
2	57	R$ 1.009.337,00
3	62	R$ 986.990,00
4	66	R$ 766.428,00
5	75	R$ 1.414.309,00
6	80	R$ 1.237.468,00
7	117	R$ 1.577.161,00
8	120	R$ 1.417.109,00
9	142	R$ 1.697.227,00
10	166	R$ 1.964.417,00
11	181	R$ 2.703.394,00
12	185	R$ 2.873.388,00
13	218	R$ 2.330.687,00
14	242	R$ 3.298.786,00
15	247	R$ 4.137.602,00
16	248	R$ 3.115.246,00
17	249	R$ 4.924.753,00
18	264	R$ 3.484.327,00
19	285	R$ 4.677.439,00
20	351	R$ 4.470.056,00
21	404	R$ 4.274.951,00
22	416	R$ 6.994.811,00
23	483	R$ 5.736.179,00
24	483	R$ 7.613.633,00
25	500	R$ 5.975.031,00



Método de Regressão Linear Simples



$$E(Y) = \alpha + \beta x_i$$

Inclinação populacional

Intercepto populacional

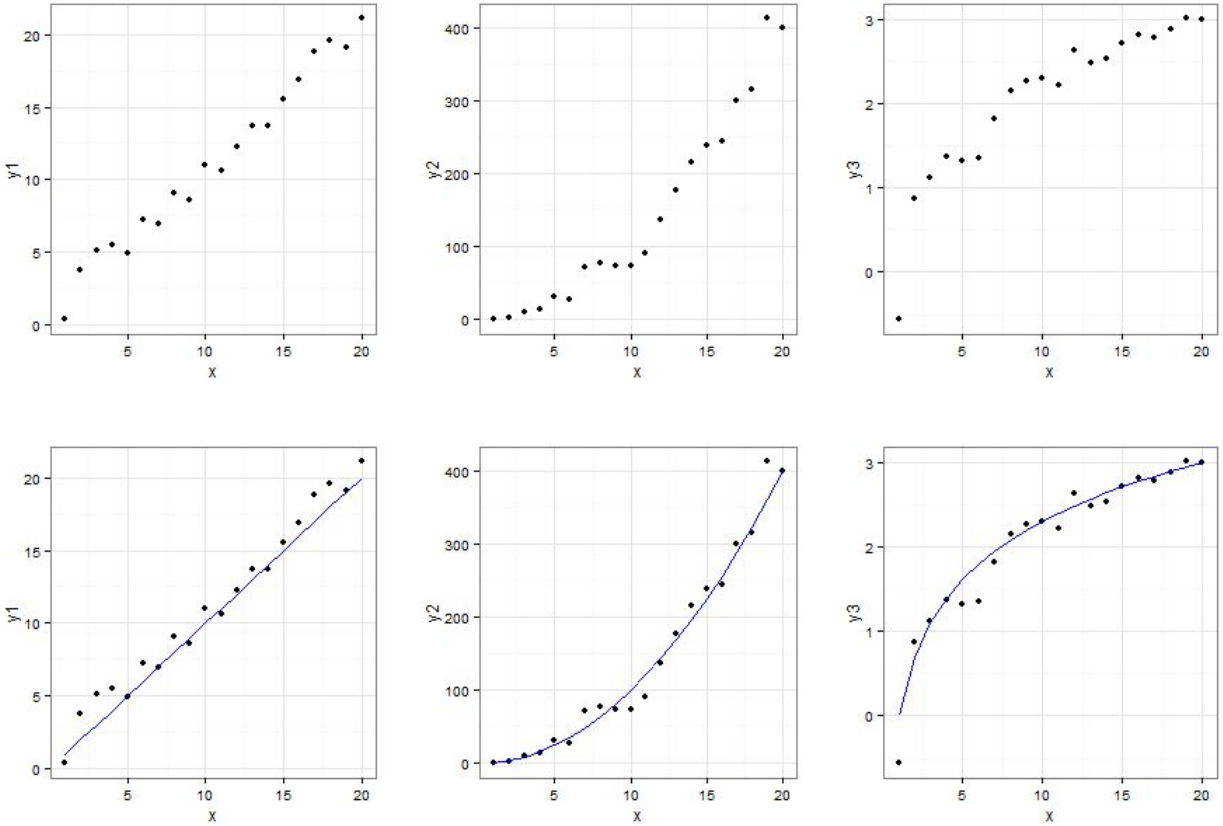
Erro Aleatório

$$Y_i = \alpha + \beta X_i + \mu_i$$

Variável Dependente

Variável Independente

Exemplos de regressões



Regressão Logística

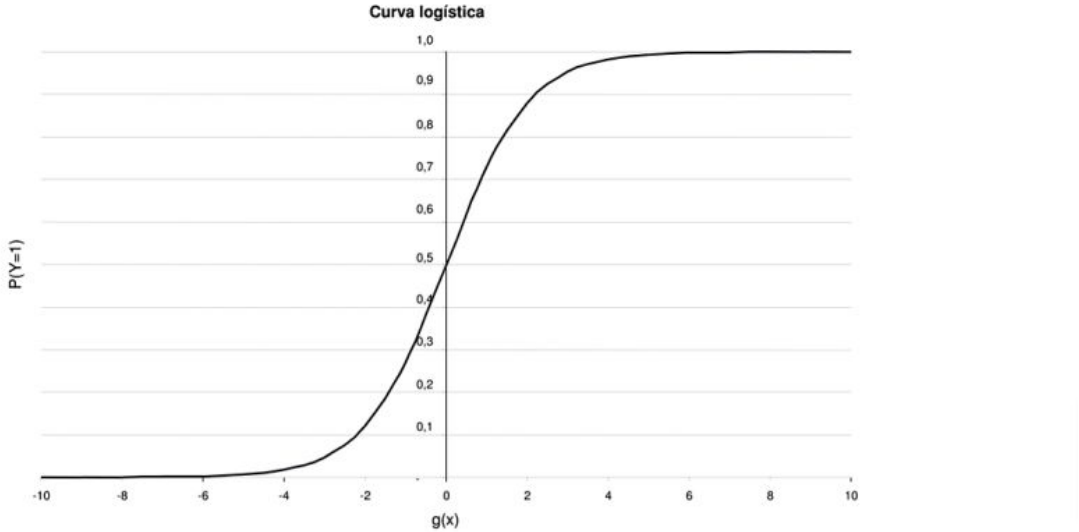
Regressão Logística

Segundo alguns pesquisadores, a Regressão Logística é uma metodologia que teve suas primeiras aplicações em 1967 em um estudo do risco de doença coronária. Essa metologia é uma das mais conhecidas nos problemas de classificação sendo aplicada até hoje em várias áreas de estudo. A regressão logística consiste em uma equação matemática que busca estimar a probabilidade de ocorrência de um determinado evento a partir de uma ou mais variáveis chamadas variáveis independentes.

Regressão Logística

O modelo de regressão logística formado por p variáveis independentes pode ser escrito na seguinte forma:

$$P(Y=1) = \frac{1}{1+e^{-g(X)}} \quad \text{onde} \quad g(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$



Regressão Logística

A equação é chamada de equação de regressão estimada, e é, essencialmente, a função que representa o objetivo do modelo de regressão logística, pois *p* é a probabilidade estimada para quaisquer valores de coeficientes e variáveis que venhamos a colocar nesta equação. Os valores dos coeficientes são obtidos pelo método de estimação da máxima verossimilhança conforme explicado na subseção seguinte.

Regressão Logística

	Vantagens	Desvantagens
Regressão Logística	• Interpretabilidade das variáveis; • Aceita variáveis categóricas e numéricas; • Baixa probabilidade de *overfitting*.	• Nem sempre os dados se ajustarão bem à curva logística; • Requer um tratamento de dados mais cuidadoso (*missings* e *outliers*).

Regressão Logística – Exemplo

A tragédia do Titanic, em 1912, é um dos desastres marítimos mais conhecidos da história. Eternizado no cinema em 1997, o navio construído em Belfast (Irlanda) naufragou quatro dias após sua viagem inaugural, que tinha como destino a cidade de Nova Iorque. Quando construído, o navio prometia ser o mais luxuoso e seguro de sua época. Entretanto, estudos posteriores indicaram falhas no sistema de segurança e evacuação. A estimativa é de 1514 mortes entre os 2224 passageiros, ou seja, aproximadamente 68% da tripulação.

Caso você estivesse no navio, qual seria a sua chance de sobrevivência?

A tragédia do Titanic é um clássico exemplo de aplicação da **Regressão Logística**. A Regressão Logística faz parte de uma família de modelos chamada Modelos Lineares Generalizados (GLM) e é adequada quando a variável de interesse (resposta) é binária, isto é, "sim" ou "não". Através da Regressão Logística é possível avaliar os fatores de influenciam a ocorrência de determinado evento.

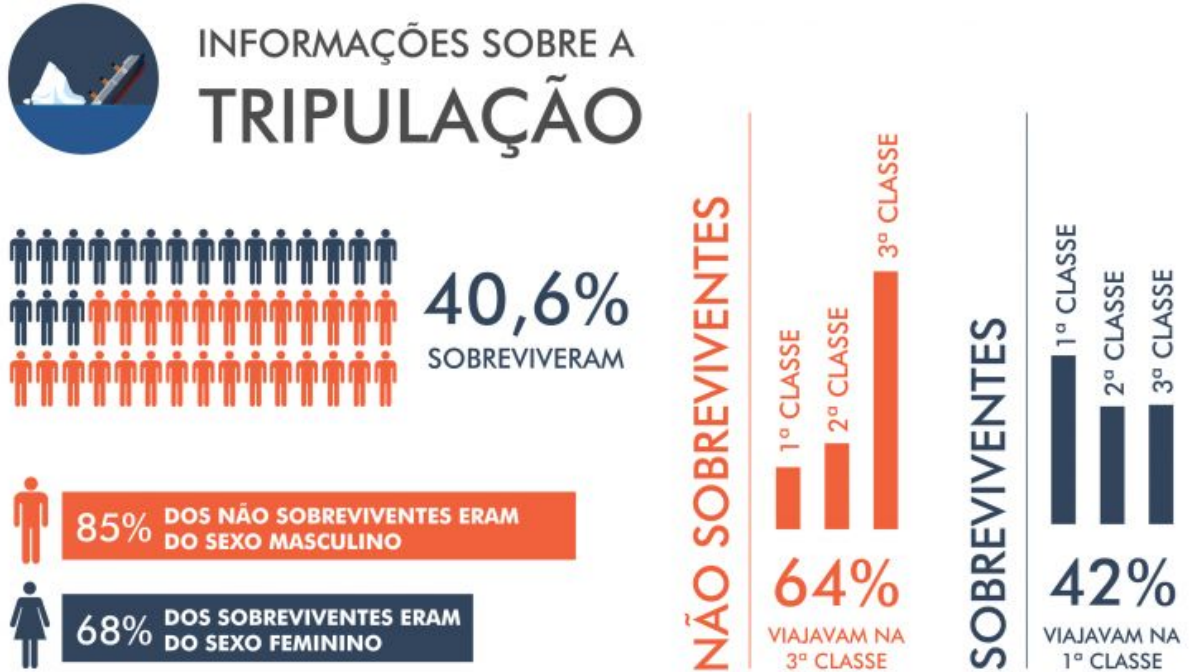
Regressão Logística – Exemplo

Características da tripulação

Na amostra, 424 indivíduos (59,38%) não sobreviveram e 290 (40,62%) sobreviveram. Cerca de 68% dos sobreviventes eram do sexo feminino e 85% dos não sobreviventes eram do sexo masculino. Em relação à classe, 42% dos sobreviventes eram da 1ª Classe e 64% dos não sobreviventes eram da 3ª Classe.

A idade média dos indivíduos que não sobreviveram foi de 30,62 anos, sendo a idade mínima de 1 ano e a máxima de 74 anos. A Idade média dos indivíduos que sobreviveram foi de 28,24 anos, sendo a idade mínima menor que 1 ano e a máxima de 80 anos.

Regressão Logística – Exemplo



Aprendizado de Máquina

Árvore de Decisão

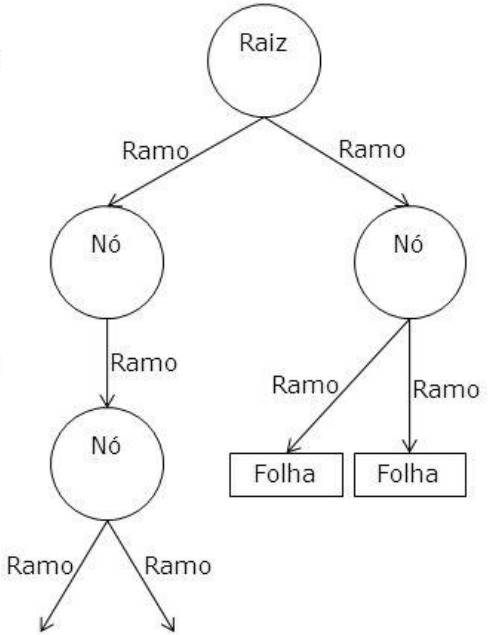
Árvore de Decisão

As Árvores de Decisão são um dos modelos mais práticos e mais usados em inferência indutiva. Este método representa funções como árvores de decisão. Estas árvores são treinadas de acordo com um conjunto de treino (exemplos previamente classificados) e posteriormente, outros exemplos são classificados de acordo com essa mesma árvore. Para a construção destas árvores são usados algoritmos como o **ID3, ASSISTANT e C4.5**.

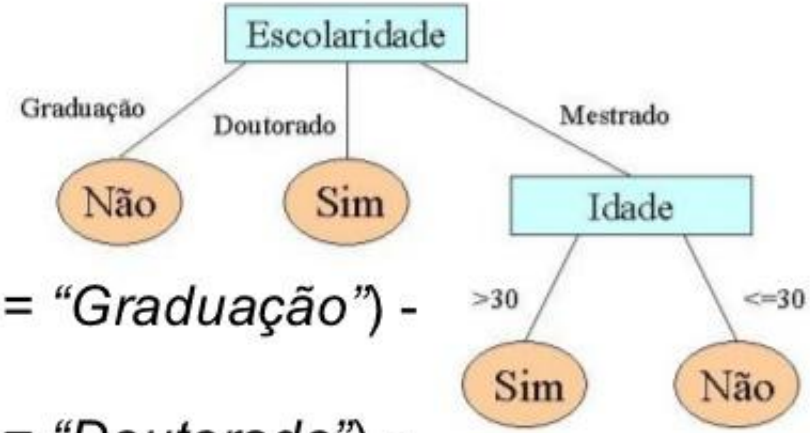
Este método de classificação pode ser facilmente compreendido através de exemplos.

Árvore de Decisão

- Cada **nó** interno da árvore corresponde a um teste do valor de uma propriedade.
- Os **ramos** dos nós são rotulados com os resultados possíveis do teste.
- Cada **nó folha** da árvore especifica o valor a ser retornado se aquela folha for alcançada.
- A representação de uma árvore de decisão é bem natural para os seres humanos.



Árvore de Decisão - Exemplo 1



1. Se (*Escolaridade = "Graduação"*) - *Rico = "Não"*
2. Se (*Escolaridade = "Doutorado"*) - *Rico = "Sim"*
3. Se (*Escolaridade = "Mestrado"*) & (*Idade = ">30"*) - *Rico = "Sim"*
4. Se (*Escolaridade = "Mestrado"*) & (*Idade = "<=30"*) - *Rico = "Não"*

Árvore de Decisão - Exemplo 2

Supondo que o objectivo é decidir se vou **Jogar Ténis**. Para tal, há que ter em conta certos parâmetros do ambiente, como o **Aspecto** do Céu, a **Temperatura**, a **Humidade** e o **Vento**. Cada um destes atributos tem vários valores. Por exemplo para a temperatura pode estar **Ameno**, **Fresco** ou **Quente**. A decisão **Sim** (ir jogar ténis) ou **Não** (não ir jogar ténis) é o resultado da classificação.

Árvore de Decisão - Exemplo 2

Para construir a Árvore de Decisão de Jogar Ténis são tidos em conta exemplos (dias) passados.

Exemplos de Treino

Dia	Aspecto	Temp.	Humidade	Vento	Jogar Ténis
D1	Sol	Quente	Elevada	Fraco	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraco	Sim
D4	Chuva	Ameno	Elevada	Fraco	Sim
D5	Chuva	Fresco	Normal	Fraco	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraco	Sim
D8	Sol	Ameno	Elevada	Fraco	Não
D9	Sol	Fresco	Normal	Fraco	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraco	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

Árvore de Decisão

Através destes exemplos é possível construir a seguinte árvore de decisão:

Árvore de Decisão

A relação entre os elementos da árvore (nós e folhas) e os atributos, valores e classificações pode ser entendida na seguinte imagem:

Árvore de Decisão

A classificação de um exemplo de acordo com a esta árvore é feita da seguinte forma:

O atributo **Aspecto** tem o valor **Sol** e a **Humidade** tem o valor **Elevada**. O exemplo é classificado com **Não**, ou seja quando esteve sol e humidade elevada não se jogou ténis. Os atributos **Temperatura** e **Vento** não são considerados, pois são desnecessário para classificar este exemplo.

Árvore de Decisão

Com Árvores de Decisão é possível representar a conjunção e disjunção de atributos. A árvore de decisão que representa a classificação para os dias em que o **Aspecto** é **Sol** e que o **Vento** está **Franco** encontra-se na seguinte figura:

Árvore de Decisão

A árvore de decisão que representa os dias em que o **Aspecto** é **Sol** ou o **Vento** está **Franco** é dada por:

Árvore de Decisão

Deve-se considera o uso de árvores de decisão em situações onde:

- As instâncias são descritas por pares atributo-valor;
- A função objeto (alvo) é de valor discreto;
- É pode ser necessário hipótese disjuntas;
- Os exemplos de treino poderão ter erro (noise);
- Faltam valores nos atributos;

Exemplos:

- - Diagnósticos médicos;
- - Análises de risco de crédito;
- - Classificação de objectos em geral

Árvore de Decisão

As árvores de decisão muito populares para problemas de classificação, regressão, análise exploratória entre outras tarefas. Características:

1. Fácil explicabilidade e interpretação, já que podemos facilmente visualizá-las (quando não são muito profundas).

1. Requerem pouco esforço na preparação dos dados, métodos baseados em árvores normalmente não requerem normalização dos dados. Além disso, conseguem lidar com valores faltantes, categóricos e numéricos.

1. Complexidade logarítmica na etapa de predição.

1. São capazes de lidar com problemas com múltiplos rótulos.

Árvore de Decisão

Problemas que podem degradar seu poder preditivo, são eles:

1. Árvore crescida até sua profundidade máxima pode decorar o conjunto de treino (o temido overfitting), o que pode degradar seu poder preditivo quando aplicado a novos dados. Isso pode ser mitigado "podando" a árvore de decisão ao atribuir uma profundidade máxima ou uma quantidade máxima de folhas.

1. São modelos instáveis (alta variância), pequena variações nos dados de treino podem resultar em árvores completamente distintas. Isso pode ser evitado ao treinarmos várias árvores distintas e agregar suas predições

1. Como vimos, o algoritmo de construção da árvore de decisão é guloso, ou seja, não garante a construção da melhor estrutura para o dados de treino em questão. Esse problema também pode ser mitigado ao treinarmos várias árvores distintas e agregar suas predições

Árvore de Decisão - Algoritmo ID3

O algoritmo ID3 (inductive decision tree) é dos mais utilizados para a construção de árvores de decisão. Este algoritmo segue os seguintes passos:

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

Coloca-se então, uma pergunta muito importante:

Como saber qual o melhor atributo a escolher?
Para lidar com esta escolha são introduzidos dois novos conceitos, a **Entropia** e o **Ganho**.

Árvore de Decisão - Algoritmo ID3

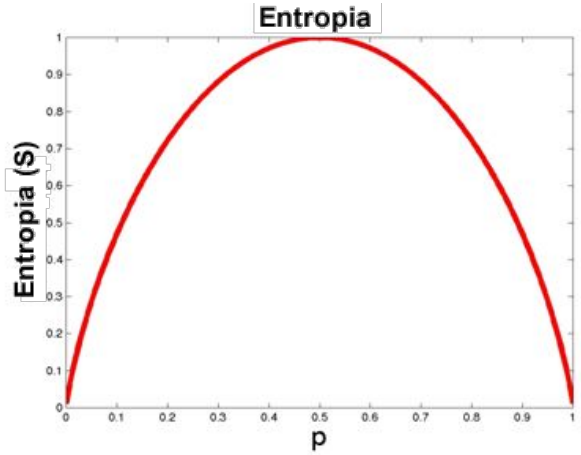
Entropia

A entropia de um conjunto pode ser definida como sendo o grau de pureza desse conjunto. Este conceito emprestado pela Teoria da Informação define a medida de "falta de informação", mais precisamente o número de bits necessários, em média, para representar a informação em falta, usando codificação óptima. Dado um conjunto *S*, com instâncias pertencentes à classe *i*, com probabilidade p_i, temos:

$$Entropia(S) = \sum p_i \log_2 p_i$$

Árvore de Decisão - Algoritmo ID3

No exemplo apenas existem duas classes de classificação, ou seja, "Jogar Ténis" (positivo, +) ou "Não Jogar Ténis" (negativo, -). Assim sendo, o valor da entropia varia de acordo com o gráfico:



Onde:

- S é o conjunto de exemplo de treino;
- p_+ é a porção de exemplos positivos;
- p_- é a porção de exemplos negativos;
- A entropia é dada pelo desdobramento da equação 1

$$Entropia(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Árvore de Decisão - Algoritmo ID3

Ganho

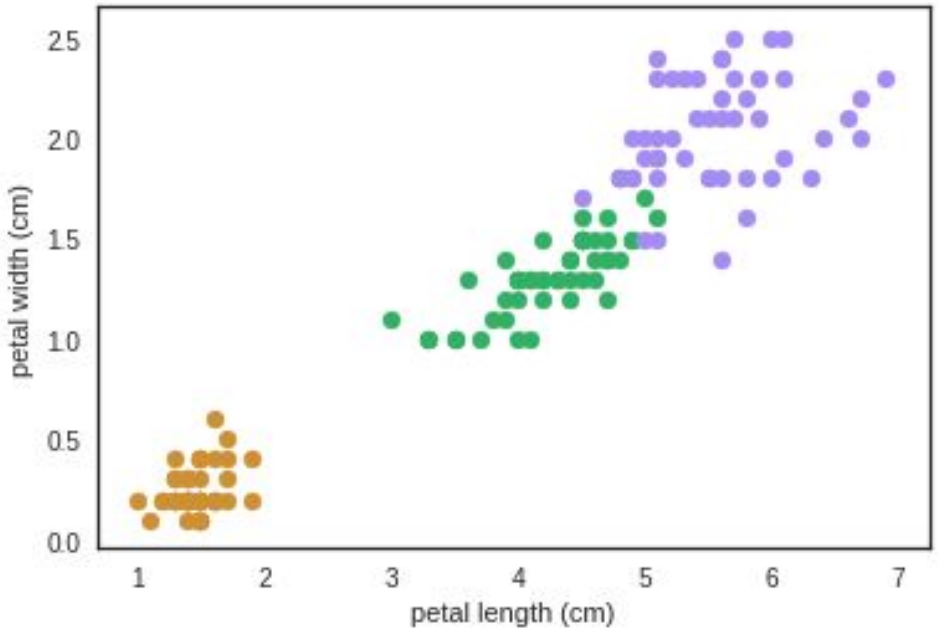
O ganho (*gain*) é define a redução na entropia. *Ganho(S,A)* significa a redução esperada na entropia de S, ordenando pelo atributo A. O ganho é dado pela seguinte equação:

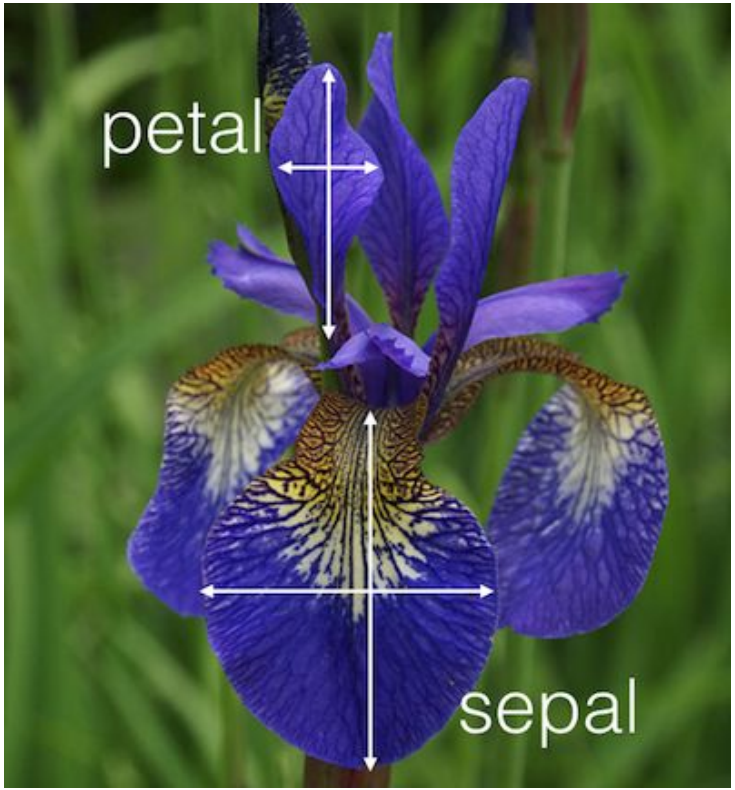
$$Ganho(S,A) = Entropia(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} \cdot Entropia(S_v)$$

Escolha do Melhor Atributo

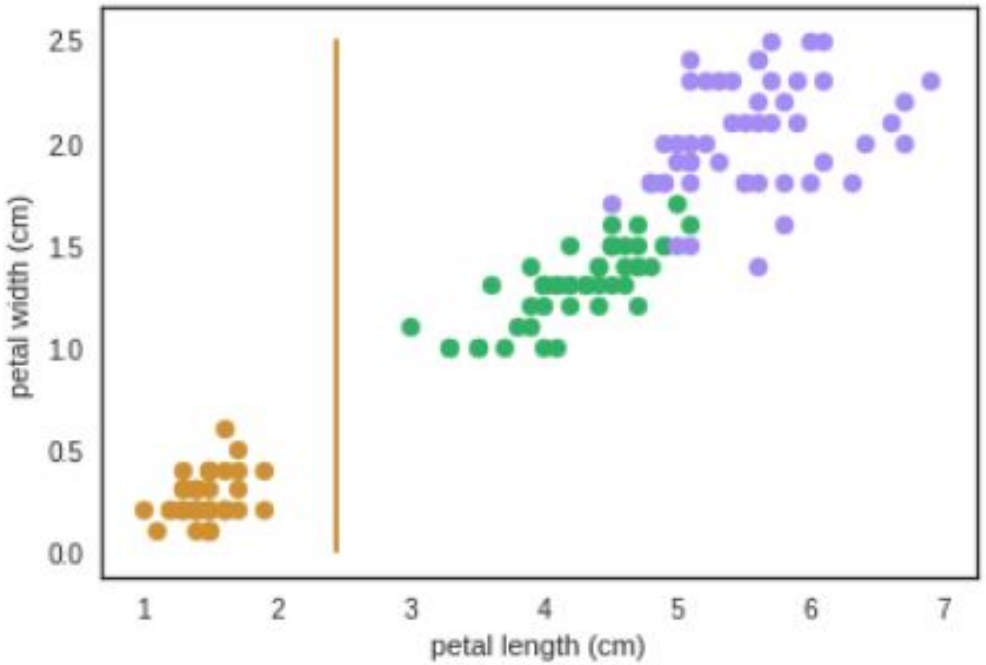
Para responder à pergunta anterior, "Como escolher o melhor atributo" é usado o ganho. Em cada iteração do algoritmo é escolhido o atributo que apresente uma maior ganho.

Árvore de Decisão - dataset Iris



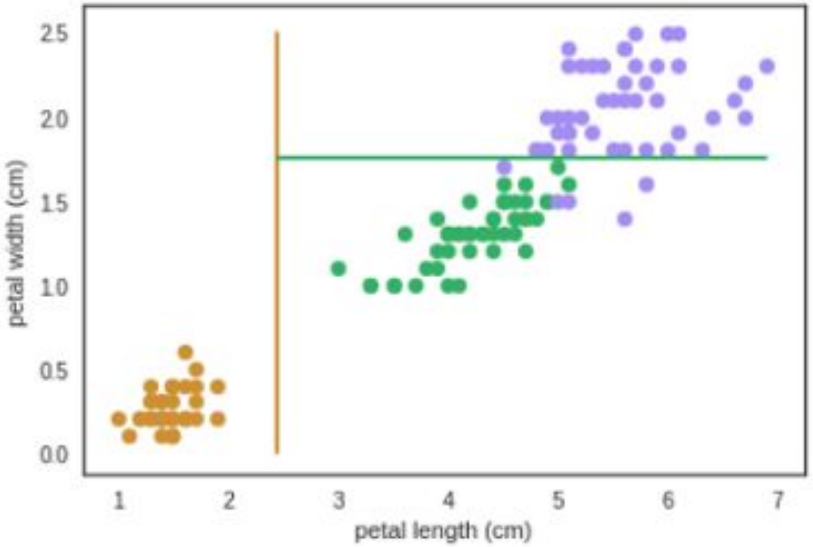


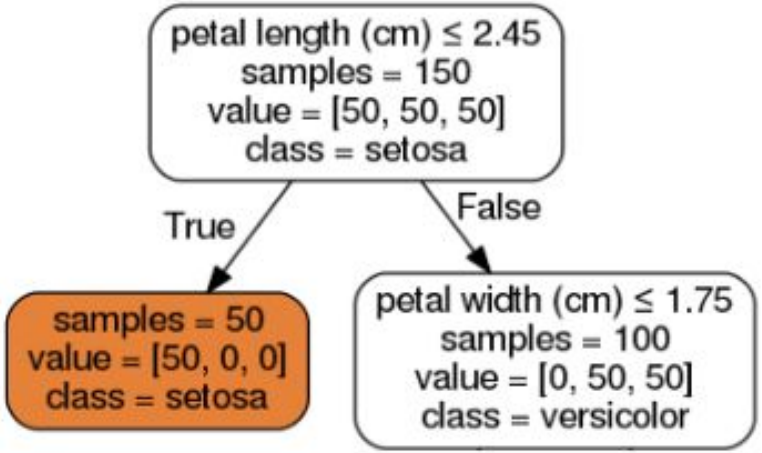
Árvore de Decisão - dataset Iris



petal length (cm) ≤ 2.45
samples = 150
value = [50, 50, 50]
class = setosa

Árvore de Decisão - dataset Iris





Árvore de Decisão - dataset Iris