Introdução ao Machine Learning [1]

Prof. Dr. Giancarlo D. Salton

agosto de 2023

[1] Universidade Federal da Fronteira Sul
Campus Chapecó
gian@uffs.edu.br

Machine Learning é a disciplina da Inteligência Artificial a qual estuda a criação de **agentes inteligentes** que aprendem através da experiência. Desta forma, ao contrário da Inteligência Artificial clássica, o agente inteligente não tem seu conhecimento programado durante sua criação: ele é **programado para aprender** a alcançar o objetivo para o qual foi desenvolvido.

Embora alguns autores definam *machine learning* como sendo o **aprendizado através de exemplos**, esta definição exclui o aprendizado por reforço positivo[2] (do Inglês *Reinforcement Learning* - RL) onde o agente aprende praticando a tarefa sem que sejam apresentadas as respostas para a sua tarefa. Neste tipo de aprendizado, o agente recebe apenas um feedback sobre o quão bem está executando aquela tarefa. A definição de aprendizado através de exemplos é mais adequada às áreas de *machine learning* supervisionado e *machine learning* não-supervisionado (ou sem supervisão).

[2] Embora sua área de estudo seja tão ampla e diversa quanto a do próprio *machine learning*, o *reinforcement learning* continua sendo uma forma de *machine learning*.

Quais os tipos básicos de Machine Learning *e onde podem ser aplicados?*

O *Machine Learning* pode ser dividido em dois grupos básicos: supervisionado; e não-supervisionado (ou sem supervisão. O aprendizado supervisionado é o tipo mais básico e é aquele no qual precisamos demonstrar para o agente inteligente o exemplo e a resposta associada àquele exemplo[3]. No aprendizado não-supervisionado, apresentamos apenas os exemplos sem incluir informações sobre o *target*[4].

Desta forma, podemos aplicar o *machine learning* para algumas tarefas básicas da mineracão de dados:

1. *Classificação*: predizer um *target* categórica. Em outras palavras, predizer a qual categoria (ou classe) dentre um conjunto finito e relativamente pequeno de categorias um determinado exemplo pertence. Podemos ainda predizer qual a probabilidade do exemplo pertencer a cada uma das categorias. Resolvido com aprendizado supervisionado.

2. *Regressão*: predizer um *target* contínua. Isso significa predizer uma determinada quantidade associada a um determinado exemplo[5]. Resolvido com aprendizado supervisionado.

[3] Imagine um professor demonstrando ao aluno informações sobre o *target* e quais são as características inclusas nas *features*. Em outras palavras, o professor supervisiona o aprendizado do aluno

[4] No aprendizado não-supervisionado, o professor apresenta os exemplos mas o aluno é livre para tirar suas próprias conclusões sobre sobre os exemplos. No entanto, isto não significa que as conclusões serão interessantes ou utilizáveis.

[5] Uma outra forma de interpretar classificação e regressão: enquanto a classificação prediz o quê vai acontecer, a regressão prediz quanto vai acontecer

3. *Combinação por similaridade*: agrupar indivíduos semelhantes com base nas *features* semelhantes com um determinado propósito. Pode ser utilizado também para classificação e regressão. Resolvido tanto com aprendizado supervisionado como com aprendizado não-supervisionado.
4. *Agrupamento*: agrupar individuos semelhantes mas sem um propósito específico. Ou seja, queremos descobrir grupos semelhantes que ocorrem naturalmente sem considerarmos um *target*. Resolvido com aprendizado não-supervisionado.
5. *Agrupamento de coocorrência*: aqui tentamos encontrar indivíduos ou itens que ocorram em situações semelhantes com base (*e.g.*, clientes que compraram X também compraram Y.). Resolvido com aprendizado não-supervisionado.
6. *Perfilamento*: determinar o comportamento característico de um indivíduo ou grupo (*i.e.*, determinar o seu "perfil"). Muito comum em detecção de anomalias ou fraudes. Resolvido com aprendizado não-supervisionado.
7. *Previsão de vínculo*: predizer se existem ou se deveriam existir ligações entre exemplos e a força desta ligação. Muito utilizado em redes sociais: "10 amigos seus conhecem X. Você gostaria de ser amigo de X"? Resolvido tanto com aprendizado supervisionado como com aprendizado não-supervisionado.
8. *Redução de dados*: substituir um conjunto grande de dados (em quantidade de exemplos ou de *features*) por um conjunto reduzido de dados que capture totalmente ou em parte as informações contidas no conjunto maior. Resolvido com aprendizado não-supervisionado.
9. *Modelagem causal*: objetivo é compreender que ações ou acontecimentos influenciam no *target* ou em algum indivíduo[6] Muito utilizado na medicina para testes de novos remédios, vacinas, tratamentos, *etc.*Resolvido com aprendizado supervisionado.

[6] De certa forma, pode ser entendido como classificação ou regressão. No entanto, vamos além e tentamos compreender os motivos das relações entre *features* e *targets* existirem. Isto pode exigir um esforço substancial para realização de experimentos controlados.

O que é o Machine Learning *"Supervisionado"?*

O *machine learning* (supervisionado) envolve o aprendizado de um modelo que descreve as relações entre um conjunto de *features* (ou *features*) e um *target* (ou *target*). Esta forma de *machine learning* gera modelos que podem ser aplicados principalmente a dois tipos de tarefas básicas de *machine learning*:

1. regressão: prever um *target* contínuo

2. classificação: prever um *target* categórico (ou a probabilidade de pertencer àquela categoria).

Devido ao fato de o modelo aprendido em *machine learning* descrever as relações entre *features* e *targets*, é necessário a obtenção de um conjunto de dados[7]) que contenha exemplos [8] com os valores corretamente preenchidos para as *features* e para o *target* associado àquele exemplo. É importante ressaltar que os modelos aprendidos refletem apenas as relações presentes nos exemplos fornecidos para o "treinamento" dos modelos. Desta forma, o maior desafio do *machine learning*, como veremos mais adiante, é aprender relações entre *features* e alvo que se estendam para além daquilo que está presente nos exemplos.

[7] Também conhecido como *dataset*.
[8] Também conhecidos de *datapoints*.

Um *dataset* é a representação a qual utilizamos para informar o treinamento de um modelo de *machine learning*, seja em formato supervisionado ou em formato não-supervisionado. Por padrão, um *dataset* segue o formato chamado ABT (*Analytics Base Table*). Neste padrão, colocamos as *features* nas colunas mais à esquerda e o *target* na coluna mais a direita caso o aprendizado seja supervisionado[9]. Além disso, é costume associar as *features* com a letra x_i sendo que i é o índice da coluna na tabela, e o *target* com a letra y. Quando nos referimos ao conjunto completo de *features*, normalmente utilizamos a letra X maiúscula. O formato da ABT é apresentado na Tabela 1.

[9] Se o aprendizado for não-supervisionado, ignoramos o fato de não haver um *target*.

features					*target*
x_1	x_2	x_3	...	x_n	y
—	—	—		—	—
—	—	—		—	—
—	—	—		—	—
—	—	—	...	—	—
—	—	—		—	—
—	—	—		—	—

Tabela 1: Formato de um *dataset*. Por padrão, *features* ficam nas colunas à esquerda enquanto o *target* é colocado na última coluna mais à direita. Cada linha do *dataset* corresponde a um *datapoint* (*i.e.*, as *features* e o *target* correspondente a um exemplo).

Como podemos observar, esta tabela lembra uma planilha de `Excel` ou a saída de uma consulta de banco de dados relacional. Embora saibamos que existam formatos mais "modernos", especialmente para armazenamento em bancos de dados, ainda utilizamos este formato mais tradicional de representação para os dados[10]. Utilizando padrão da ABT fica mais fácil para aplicarmos técnicas de *machine learning* sobre o conjunto de dados.

[10] Uma área de pesquisa que vem ganhando força nos últimos anos dentro do *machine learning* é a área que explora diferentes formatos para a representação dos dados. Embora muito esforço esteja sendo colocado nesta área e existirem várias demonstrações sobre o quão promissora ela é, ainda estamos longe de conseguirmos utilizar outros formatos de representação de dados com eficiência. Isso é ainda mais perceptível com grandes quantidades de dados que requerem tecnologias como `Hadoop` e `Spark`.

A Tabela 2 apresenta um exemplo de um dataset real. Esta é uma amostra do *dataset* Iris publicado em 1953 e que até hoje representa um caso típico de testes para modelos de *machine learning*.

ID	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	SPECIES (*target*)
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	7.0	3.2	4.7	1.4	versicolor
5	6.4	3.2	4.5	1.5	versicolor
6	6.9	3.1	4.9	1.5	versicolor
7	6.3	3.3	6.0	2.5	virginica
8	5.8	2.7	5.1	1.9	virginica
9	7.1	3.0	5.9	2.1	virginica
	x_1	x_2	x_3	x_4	y

Tabela 2: Amostra do clássico *dataset* "Iris" para demonstrar um exemplo de *dataset* com exemplos reais. O "Iris" foi introduzido por Ronald Fisher em 1936 e tornou-se um caso de teste típico para algoritmos de *machine learning*. O *dataset* é constituído por quatro *features* (SEPAL LENGTH, SEPAL WIDTH, PETAL LENGTH e PETAL WIDTH) e um *target* (SPECIES).

Features (variáveis descritivas) e *Target* (variável alvo) são, respectivamente, o conjunto de informações e a resposta correta relativa a um exemplo do domínio. Um "exemplo", "observação", *datapoint* ou *indivíduo* é a a combinação de features + target. Uma *query* é um exemplo para o qual queremos predizer o *target* correto (i.e., uma *query* é uma observação sem um *target* associado). O *dataset* (ou conjunto de exemplos) é conjunto de observações de um determinado domínio (problema). Um algoritmo é processo utilizado para aprender um modelo. O modelo é o conjunto de "regras" ou "relações" aprendidas. *No contexto da Inteligência Artificial e* machine learning*, o modelo é o Agente Inteligente!*

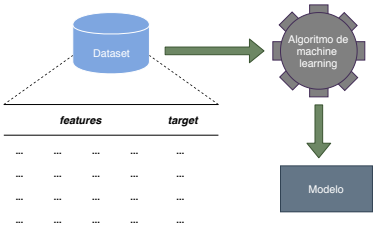


Figura 1: Ilustração do processo de aprendizado: um dataset formado por exemplos é fornecido a um algoritmo de *machine learning* que, por sua vez, itera sobre os exemplos e gera um modelo a partir das relações entre *features* e *target*.

A Figura 1 ilustra o processo de "aprendizado" de *machine learning*: um *dataset* contendo *features* e suas respectivas *targets* é fornecido como entrada para um algoritmo que itera sobre os exemplos e, por sua vez, gera um modelo das relações entre *features* e *target*. No processo de predição, depois de gerado o modelo, podemos pegar uma *query* e obter uma predição para determinar qual a resposta (*i.e.*, o *target*) correta para aquela instância. A Figura 2 ilustra o processo de predição. Vale ressaltar que a resposta retornada como saída é baseada nas relações que o modelo aprendeu[11] e que, nem sempre, são 100% corretas[12].

Figura 2: Ilustração do processo de predição: uma nova instância, sem informação sobre o *target* é fornecida ao modelo gerado durante o aprendizado. O modelo, por sua vez, apresenta como saída o valor do *target* correspondente àquela instância.

A Figura 1 ilustra o processo de "aprendizado" de *machine learning*: um *dataset* contendo *features* e suas respectivas *targets* é fornecido como entrada para um algoritmo que itera sobre os exemplos e, por sua vez, gera um modelo das relações entre *features* e *target*. No processo de predição, depois de gerado o modelo, podemos pegar uma *query* e obter uma predição para determinar qual a resposta (*i.e.*, o *target*) correta para aquela instância. A Figura 2 ilustra o processo de predição. Vale ressaltar que a resposta retornada como saída é baseada nas relações que o modelo aprendeu[13] e que, nem sempre, são 100% corretas[14].

[11] Se um exemplo de relação (*i.e.*, um tipo de *target*) não está presente no conjunto de dados, o modelo será incapaz de aprendê-la.

[12] Como veremos mais adiante, um modelo que está 100% correto, nem sempre é um bom modelo.

[13] Se um exemplo de relação (*i.e.*, um tipo de *target*) não está presente no conjunto de dados, o modelo será incapaz de aprendê-la.

[14] Como veremos mais adiante, um modelo que está 100% correto, nem sempre é um bom modelo.

Se analisarmos a Tabela 3 com cuidado, observaremos que existe uma relação simples entre as *features* e o *target* e que pode ser representada através da regra (`Python`) em 15.

Para entendermos melhor o que são as relações entre *features* e *target*, vamos analisar o exemplo a seguir. A Tabela 3 contém um *dataset* com 3 *features* e um *target* (Classe).

ID	Profissão	Idade	Proporção Salário-Empréstimo	Classe
1	indústria	34	2.96	pago
2	autônomo	41	4.64	atraso
3	autônomo	36	3.22	atraso
4	autônomo	41	3.11	atraso
5	indústria	48	3.80	atraso
6	indústria	61	2.52	pago
7	autônomo	37	1.50	pago
8	autônomo	40	1.93	pago
9	indústria	33	5.25	atraso
10	indústria	32	4.15	atraso

Tabela 3: Exemplo de *dataset*, contendo as *features* Profissão, Idade e Proporção Salário-Empréstimo, e o *target* Resultado.

Se analisarmos a Tabela 3 com cuidado, observaremos que existe uma relação simples entre as *features* e o *target* e que pode ser representada através da regra (`Python`) em 15.[15]

A partir deste exemplo de um modelo que poderia ser utilizado para fazer predições, podemos observar algumas coisas importantes sobre *features* no *machine learning*. Primeiro, nem todas as *features* foram utilizadas no "modelo". Segundo, a variável descritiva Proporção Salário-Empréstimo não é uma variável que normalmente encontramos num banco de dados[16]. O design e a seleção de variáveis são dois temas recorrentes dento do *machine learning* conforme estudaremos mais adiante.

Embora este exemplo demonstre alguns conceitos interessantes sobre o *machine learning*, seu valor realmente aparece quando processamos conjuntos de dados que possuem muitos exemplos compostos por muitas *features*. Um exemplo pode ser encontrado na Tabela 4:

Mais uma vez, se analisarmos com cuidado, podemos extrair uma regra mais complexa que descreve a relação entre as *features* e o *target*, conforme demonstrado na regra em [17]

Note como foi mais difícil extrair as regras de forma manual sem olharmos a resposta dada. É nestes casos mais complexos que o *machine learning* demonstra sua utilidade. No entanto, é sempre bom ressaltar que a qualidade das regras extraídas é ligada diretamente com a qualidade dos dados fornecidos para o algoritmo. Iremos retor-

[15]

```
if 'Proporção Salário-Empréstimo' > 5:
    classe = 'atraso'
else:
    classe = 'pago'
```

[16] Isto não significa que jamais encontraremos estes tipos de variáveis num banco de dados. Muitas vezes, engenheiros de software apenas armazenam dados brutos como valores de salário e empréstimo e realizam cálculos de proporções apenas quando uma funcionalidade específica do software é acionada.

[17]

```
if 'Proporção Salário-Empréstimo' < 1.5:
    classe = 'pago'
elif 'Proporção Salário-Empréstimo' > 4:
    classe = 'atraso'
elif 'Idade' < 40
   and 'Profissão' == 'indústria':
    classe = 'atraso'
else:
    classe = 'pago'
```

ID	Valor	Salário Anual	Proporção Salário-Empréstimo	Idade	Profissão	Propriedade	Classe
1	245,100	66,400	3.69	44	indústria	fazenda	pago
2	90,600	75,300	1.2	41	indústria	fazenda	pago
3	195,600	52,100	3.75	37	indústria	fazenda	atraso
4	157,800	67,600	2.33	44	indústria	apto.	pago
5	150,800	35,800	4.21	39	autônomo	apto.	atraso
6	133,000	45,300	2.94	29	indústria	fazenda	atraso
7	193,100	73,200	2.64	38	autônomo	casa	pago
8	215,000	77,600	2.77	17	autônomo	fazenda	pago
9	83,000	62,500	1.33	30	autônomo	casa	pago
10	186,100	49,200	3.78	30	indústria	casa	atraso
11	161,500	53,300	3.03	28	autônomo	apto.	pago
12	157,400	63,900	2.46	30	autônomo	fazenda	pago
13	210,000	54,200	3.87	43	autônomo	apto.	pago
14	209,700	53,000	3.96	39	indústria	fazenda	atraso
15	143,200	65,300	2.19	32	indústria	apto.	atraso
16	203,000	64,400	3.15	44	indústria	fazenda	pago
...	...	...	...	...	...	...	...

Tabela 4: Exemplo de *dataset* mais complexo, contendo as *features* Valor, Salário-Anual, Proporção Salário-Empréstimo, Idade, Profissão e Propriedade, e o *target* Resultado.

nar a este ponto sobre a qualidade dos dados em outra aula.

Como funciona o Machine Learning*?*

Todos os algoritmos de *machine learning* possuem um forma básica de funcionamento que pode ser reduzida a alguns pontos. Primeiro ponto: os algoritmos iteram sobre os dados em busca das relações entre *features* e *target*. Uma forma de se interpretar esta ação, é que os algoritmos de *machine learning* constroem um modelo que se "ajustam" aos dados[18]. Assim, conforme dito anteriormente, os modelos aprendidos refletem apenas as relações presentes nos exemplos fornecidos para o seu "treinamento". Este tipo de aprendizado também pode ser entendido como um processo de "otimização": começamos com um modelo genérico e, conforme as iterações se segue, o modelo será otimizado para o conjunto de exemplos utilizados para o treinamento.

O segundo ponto é que os algoritmos são capazes de extrair apenas relações presentes dentro do conjunto de exemplos utilizados para o aprendizado. Isto significa dizer que se criarmos um modelo de *machine learning* para predizer se no próximo dia irá chover ou se fará Sol (*i.e.*, nosso *target* $\in$ {'chuva', 'sol'}), ele não será capaz de predizer se o dia estará nublado. Deste modo, precisamos que o conjunto de dados de treinamento contenha exemplos de todos os tipos de predi-

[18] O *Machine Learning* pode ser interpretado como *curve-fitting*: se criarmos um gráfico dos exemplos antes do processo de treinamento e compararmos com um gráfico do modelo aprendido, observaremos que os gráficos terão um formato bastante próximo um do outro.

ções (relações) as quais desejamos que o modelo capture[19].

Assim, uma forma simples e intuitiva de realizar o aprendizado, seria uma busca dentre um conjunto possivelmente infinito de modelos para encontrar aquele modelo que melhor capture as relações entre *features* e *target*. O problema neste caso é como fazer busca, afinal estamos considerando um conjunto potencialmente infinito de possibilidades.

Um critério para busca poderia ser consideremos apenas os modelos que são consistentes com os dados. Um modelo consistente pode ser entendido como um modelo que é capaz de acertar todas as predições para os exemplos utilizados em seu treinamento. No entanto, o *machine learning* é considerado um problema *mal-posto*. Em outras palavras, um problema mal-posto é um problema cujas informações para resolvê-lo não estão todas presentes em seu enunciado. No caso do *machine learning*, nem sempre temos todos os exemplos para que o modelo aprenda as relações entre variável descritiva e *target*[20].

Vamos ilustrar este problema com um conjunto de dados de exemplo. A Tabela 5 apresenta um *dataset* contendo exemplos de clientes de um supermercado representados por três *features* binárias (Filhos, Álcool e Orgânicos), e um *target* que corresponde ao tipo de grupo familiar aos quais os clientes pertencem ('casal', 'família' ou 'solteiro').

ID	Filhos	Álcool	Orgânicos	Grupo
1	não	não	não	casal
2	sim	não	sim	família
3	sim	sim	não	família
4	não	não	sim	casal
5	não	sim	sim	solteiro

Tabela 5: Exemplo de dataset contendo informações sobre clientes de um supermercado e *features* binárias Filhos, Álcool e Orgânicos, descrevendo se estes clientes compram determinados tipos de produto, e o *target* Grupo.

Perceba que estamos lidando com *features* binárias que podem receber apenas valores 'Sim' ou 'Não', o que nos dá um total de 8 (*i.e.*, 2^3) combinações possíveis para os valores das *features*. Vamos supor, no entanto, que por alguma razão não especificada,conseguimos coletar apenas os cinco exemplos da Tabela 5.

Podemos facilmente facilmente gerar todas as combinações possíveis para os valores das *features*: {'Não', 'Não', 'Não' }; {'Não', 'Não', 'sim' }; {'Não', 'Sim', 'Não' }; *etc.*, sem considerarmos os valores do *target*. Estas combinações estão demonstradas na Tabela 6.

Vamos agora considerar os valores do *target* junto aos valores das *features*: se começarmos a atribuir aleatoriamente valores para o *target*, vamos perceber que existem 6.561 combinações possíveis ao todo! Demonstramos uma parte destas combinações na Tabela 7.

[19] Uma interpretação dos modelos gerados por *machine learning*, válida especialmente para os modelos aplicados em classificação, é que estes modelos são apenas "rotuladores de exemplos".

[20] No caso do *machine learning*, os exemplos são as informações e as *features* e *target* compõem o enunciado.

ID	Filhos	Álcool	Orgânicos	Grupo
1	não	não	não	?
2	não	não	sim	?
3	não	sim	não	?
4	não	sim	sim	?
5	sim	não	não	?
6	sim	não	sim	?
7	sim	sim	não	?
8	sim	sim	sim	?

Tabela 6: Todas as possíveis combinações para as *features* binárias Filhos, Álcool e Orgânicos sem considerar o valor do *target*.

Flh	Álc	Org	Grp	$\mathbb{M}_1$	$\mathbb{M}_2$	$\mathbb{M}_3$	$\mathbb{M}_4$	$\mathbb{M}_5$	...	$\mathbb{M}_{6561}$
não	não	não	?	casal	casal	solteiro	casal	casal		casal
não	não	sim	?	solteiro	casal	solteiro	casal	casal		solteiro
não	sim	não	?	família	família	solteiro	solteiro	solteiro		família
não	sim	sim	?	solteiro	solteiro	solteiro	solteiro	solteiro		casal
sim	não	não	?	casal	casal	família	família	família	...	família
sim	não	sim	?	casal	família	família	família	família		casal
sim	sim	não	?	solteiro	família	família	família	família		solteiro
sim	sim	sim	?	solteiro	solteiro	família	família	casal		família

Tabela 7: Amostra das possíveis combinações quando combinamos os valores das *features* binárias Filhos, Álcool e Orgânicos com os valores possíveis do *target*.

Se olharmos atentamente para a tabela, perceberemos que algumas destas combinações não condizem com as cinco respostas coletadas na Tabela 5. Estas combinações são "inconsistentes" com os exemplos coletados. Na Tabela 8, excluímos algumas das combinações visíveis que não são consistentes com os dados, tornando-as sombreadas.

Mesmo excluindo algumas das combinações inconsistentes, ainda temos pelo menos três combinações que são consistentes com os exemplos da Tabela 5. Dentre as combinações consistentes, como escolher o modelo correto? Este exemplo nos mostra que o *machine learning* é realmente um problema "mal-posto" pois não conseguimos encontrar uma solução apenas no enunciado inicial (*i.e.*, o conjunto de exemplos da Tabela 5).

Uma outra forma de entendermos a consistência é pensar em memorização. Muito embora não sejam a mesma coisa, um modelo consistente pode ser comparado a uma "memória" dos exemplos. No entanto, muitas vezes temos que lidar com ruídos nos valores das *features*, dados faltantes, *etc.*Quando isso acontece, uma "memorização" do conjunto de exemplos não é desejável, pois as relações aprendidas também irão conter estes mesmos ruídos. Assim, o objetivo ao treinar um modelo de *machine learning* é que este modelo seja capaz de "generalizar" as relações aprendidas para além dos exemplos utilizados durante o treino e que seja capaz de ignorar o ruído contido naqueles exemplos.

FLH	ÁLC	ORG	GRP	$\mathbb{M}_1$	$\mathbb{M}_2$	$\mathbb{M}_3$	$\mathbb{M}_4$	$\mathbb{M}_5$	...	$\mathbb{M}_{6\,561}$
não	não	não	?	casal	casal	solteiro	casal	casal		casal
não	não	sim	?	solteiro	casal	solteiro	casal	casal		solteiro
não	sim	não	?	família	família	solteiro	solteiro	solteiro		família
não	sim	sim	?	solteiro	solteiro	solteiro	solteiro	solteiro		casal
sim	não	não	?	casal	casal	família	família	família	...	família
sim	não	sim	?	casal	família	família	família	família		casal
sim	sim	não	?	solteiro	família	família	família	família		solteiro
sim	sim	sim	?	solteiro	solteiro	família	família	casal		família

Tabela 8: Amostra das possíveis combinações quando combinamos os valores das *features* binárias FILHOS, ÁLCOOL e ORGÂNICOS com os valores possíveis do *target*. Combinações inconsistentes estão sombreadas.

O *machine learning* precisa de um guia pois, apenas observando os modelos consistentes, qualquer um daqueles modelos poderia ser o escolhido durante a busca. A busca pelo melhor modelo é guiada pelo conjunto de dados de treino e pelo que chamamos de viés indutivo. O viés indutivo que nada mais é do que um conjunto de suposições necessárias para se definir um critério de seleção de um modelo de *machine learning*.

O viés indutivo é composto por dois vieses: o viés de restrição; e o viés de preferência. Embora pareçam complicados, são dois conceitos bem simples:

i. Bias *de restrição*: é relacionado ao tipo de modelo o qual colocamos o *machine learning* aprender. Em outras palavras, o viés de restrição é o algoritmo que utilizamos para aprender o modelo. Por exemplo, se utilizarmos o algoritmo ID3, este algoritmo irá aprender uma estrutura em forma de árvore. Assim, este algoritmo não pode aprender um modelo com estrutura rede neural artificial. Assim, ao selecionarmos o ID3, estamos *restringindo* o modelo que será aprendido - neste caso restringindo à estrutura de árvore.

ii. Bias *de preferência*: é relacionado aos parâmetros de cada algoritmo e, por isso, é dependente de cada um. Por exemplo, podemos definir que o ID3 irá aprender uma estrutura de árvore com no máximo 5 níveis. Assim, estamos dando preferência para estruturas de árvore com profundidade entre 1 e 5 níveis. Portanto, o algoritmo não irá aprender modelos 6 níveis ou mais. Neste ponto é necessário entender como cada algoritmo funciona para selecionar corretamente os seus parâmetros[21] e aprender o melhor modelo possível para o conjunto de exemplos fornecido.

[21] Iremos estudar os detalhes de alguns dos principais algoritmos de *machine learning* nas próximas aulas

O quê pode dar errado no Machine Learning*?*

AGORA QUE SABEMOS DE forma geral como funciona o "aprendizado" dos modelos, podemos observar as situações em que a busca pode falhar e gerar modelos incorretos ou sub-ótimos. Primeiro, devemos considerar é o fato de que "não existe almoço grátis": se um algoritmo gerou um modelo com boa performance para um conjunto de exemplos, não significa que irá funcionar em outro conjunto de exemplos, mesmo que sejam parecidos. Para piorar, o mesmo algoritmo pode não apresentar a mesma performance ainda que utilizemos uma quantidade de exemplos igual com as mesmas *features*. Lembre-se que sempre trabalhamos com uma amostra de dados, independente de quantos exemplos tenhamos coletados. Por ser uma amostra, os dados serão naturalmente diferentes[22]. Além disso, um modelo de *machine learning* sempre será uma representação simplificada da realidade, seja por limitações na coleta de exemplos de treino ou nas *features* coletadas, seja por questões de tratabilidade e capacidade computacional.

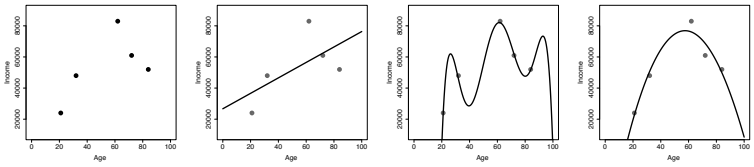
Temos que tomar cuidado também com o viés indutivo. Se escolhermos o algoritmo errado, o modelo gerado com certeza será incorreto ou inadequado. Mesmo se escolhermos o algoritmo certo mas escolhermos os parâmetros incorretamente, o modelo provavelmente também será incorreto ou inadequado. Quando falamos em incorreto ou inadequado, estamos falando em dois dos problemas principais que ocorrem no *machine learning*: *overfitting* ("sobreajuste"); e *underfitting* ("subajuste"). Ambos os problemas tem seu nome derivado da interpretação de que um modelo de *machine learning* é "ajustado" iterativamente sobre um conjunto de exemplos[23]. Observe a Figura 3: ela é a representação gráfica dos dados contidos na Tabela 9.

[22] Imagine que trabalhemos com exemplos de clientes de dois bancos diferentes. Mesmo que coletemos exemplos apenas daqueles clientes que possuem contas em ambos os bancos, é provável que seus hábitos de utilização da conta em cada um dos bancos varie: em um banco o cliente pode receber seu salário, no outro banco pode possuir cartão de crédito *etc.*Mesmo que os cliente receba salário em ambas as contas, serão valores de empregos diferentes. Mesmo que o cliente utilize cartão de crédito em ambos os bancos, suas transações (compras) serão diferentes.

[23] Conceito de *curve-fitting*

ID	AGE	INCOME
1	21	24.000
2	32	48.000
3	62	83.000
4	72	61.000
5	84	52.000

Tabela 9: Conjunto de dados contendo exemplos para a variável descritiva AGE e o sua respectivo *target* INCOME.



(a) Dataset (b) *Underfitting* (c) *Overfitting* (d) Aproximado

Figura 3: Encontrando o equilíbrio entre complexidade e simplicidade do modelo (*i.e.*, entre *underfitting* e *overfitting*) ao tentar predizer um *target* INCOME a partir da *feature* AGE.

A Figura 3 ilustra os conceitos de *overfitting* e *underfitting*. Na Figura 3(a) podemos perceber a partir do conjunto de dados (*dataset*), que os pontos que representam os exemplos no gráfico formam uma figura que lembra uma "parábola". Na Figura 3(b), temos a representação de um modelo aprendido que possui um formato de reta[24]: fica óbvio que a reta não é adequada aos dados que possuem um formato de parábola[25]. Já ná Figura 3(c), temos um gráfico em formato de curva que lembra um pouco uma montanha-russa, mas que passa por todos os pontos mostrados no gráfico. Embora este modelo pareça ter uma boa performance, devemos analisar se este formato do modelo é realmente o caso dos exemplos demonstrados na Figura 3(a). Inicialmente tínhamos a expectativa de que um modelo parecido com uma parábola poderia se ajustar aos exemplos, mas na Figura 3(c) obtivemos um modelo muito mais complexo do que o esperado. Se usarmos nossa imaginação e formos criteriosos, perceberemos que os locais onde o gráfico desceu (AGE $\approx$ 40) ou subiu repentinamente (AGE $\approx$ 90), são locais onde os valores dos *target* são incorretamente preditos. Faz mais sentido que os valores do *target* mantivessem o formato de parábola[26]. Neste caso da Figura 3(c), o modelo provavelmente está em *overfitting*, pois embora se ajuste perfeitamente aos dados, entendemos que em alguns pontos ele irá predizer de forma incorreta os valores do *target*. Figura 3(d) demonstra um modelo que se aproxima bastante do formato esperado de parábola para o conjunto de exemplos mas, que mesmo assim, deixa "escapar" alguns exemplos. Além disso, ele demonstra outro ponto importante do *machine learning*: qualquer modelo gerado sempre será apenas uma abstração da realidade, contida no conjunto de exemplos e nas *features* utilizadas para o aprendizado. Além disso, devemos levar em consideração o fato de que sempre trabalhamos com uma amostra dos dados e nunca com todas combinações (possivelmente infinitas) de valores para as *features*. Mais ainda, temos que considerar que, por mais que limpemos os dados para remover ruídos, quase sempre há alguma coisa que escape desta limpeza, seja por serem valores válidos ou por erros de limpeza.

Estes dois problemas são difíceis de detectar pois podem ocorrer por diversos motivos que nem sempre são óbvios. O *overfitting* acontece quando o modelo aprendido representa muito bem as relações entre *features* e *target* mas não consegue predizer corretamente o valor do *target* para qualquer instância de fora dos exemplos de treino. Este tipo de problema pode ocorrer por diversas razões: o algoritmo utilizado pode aprender um modelo que é complexo demais para o conjunto de exemplos; podemos ter escolhido os parâmetros do algoritmo (viés de restrição) de forma incorreta e isto tornou o modelo muito complexo; podemos ter deixado a iteração de treino continuar por tempo demais; as variáveis são incorretas; temos muito ruído nos

[24] Provavelmente este modelo foi aprendido com o algoritmo de Regressão Linear. Iremos ver um pouco mais sobre este algoritmo nas próximas aulas.

[25] Esta comparação entre gráficos que chamamos de *curve-fitting*.

[26] Ainda mais se levarmos em consideração o que acontece na vida real: pessoas mais jovens, em início de carreira tendem a ganhar menos (o seu *income* é baixo); pessoas com mais experiência, próximas do auge de suas carreiras, tendem a ganhar; pessoas aposentadas tendem novamente a ter uma diminuição na sua renda.

valores das *features*; entre outras. O *underfitting*, por sua vez, acontece quando o modelo aprendido não representa as relações entre *features* e *target* de forma correta: os exemplos utilizados no treinamento não são representativos o suficiente em relação ao problema; o algoritmo escolhido não é complexo o suficiente; os parâmetros escolhidos tornaram o modelo muito simples; não temos *features* ou exemplos o suficiente; não deixamos a iteração do algoritmo continuar por tempo suficiente; *etc.*O desafio é descobrir a causa destes problemas e agir sobre elas. A avaliação dos modelos resultantes do *machine learning* é uma área importante do processo e será estudada em separado.

Dentro do *machine learning*, existem vários algoritmos que podem ser utilizados para aprender modelos que representam as relações entre *features* e *target*. No entanto, todos eles podem ser classificados em uma de quatro "famílias" básicas:

i. Algoritmos baseados em informação;

ii. Algoritmos baseados em similaridade;

iii. Algoritmos baseados em probabilidade; e

iv. Algoritmos baseados em erro.

Estudaremos com um pouco mais de detalhe todas estas famílias de algoritmos nas próximas aulas.

Resumo

O *machine learning* supervisionado é utilizado para aprender de forma automática as relações entre *features* e um *target*, contidas em um conjunto de exemplos. Os algoritmos funcionam buscando o melhor modelo que represente estas relações, dentre um número possivelmente infinito de possibilidades. O que guia esta busca pelo melhor modelo é o conjunto de exemplos utilizados para o aprendizado e o viés indutivo. O viés indutivo é composto pelo viés de restrição (algoritmo) e pelo viés de preferência (parâmetros do algoritmo). O objetivo principal do aprendizado é gerar um modelo que seja capaz de aprender para além dos dados utilizado[27] em seu treinamento e que seja capaz de fazer previsões corretas para instâncias que nunca foram vistas durante o treinamento.

[27] Também chamado de generalização para além dos dados de treino.

O conjunto de dados utilizados em treinamento é sempre uma amostra de todas as possíveis combinações entre os valores das *features*. Por isso, consideramos o *machine learning* como sendo "problema mal-posto" e, também, consideramos que "não há almoço grátis". Estes fatos podem acarretar em dois tipos de problemas: *underfitting*,

que ocorre quando o modelo aprendido é simples demais para o conjunto de exemplos; e o *overfitting*, que ocorre quando o modelo possui uma boa performance sobre os exemplos de treino, mas prediz incorretamente instâncias que não estavam presentes no treino. A busca pelo equilíbrio entre *overfitting* e *underfitting* é um dos temas centrais na área de *machine learning*.

Referências

Katti Faceli, Ana Carolina Lorena, João Gama, Tiago Agostinho de Almeida, and André Carlos Ponce de Leon Ferreira de Carvalho. *Inteligência Artificial Uma Abordagem de Aprendizado de Máquina*. LTC, 2 edition, 2021.

John D. Kelleher, Brian Mac Namee, and Aoife D'Arcy. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples and Case Studies*. MIT Press, 2 edition, 2020.

Foster Provost and Tom Fawcett. *Data Sience para Negócios*. MIT Press, 1 edition, 2016.

Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 3 edition, 2016.