João Victor Barbosa Alves

Seleção Incremental de Variáveis para Aprendizado de Máquina utilizando Preditor Linear e Validação Cruzada

Belo Horizonte, Minas Gerais - Brasil

2018

João Victor Barbosa Alves

Seleção Incremental de Variáveis para Aprendizado de Máquina utilizando Preditor Linear e Validação Cruzada

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus lacinia, erat vitae pulvinar malesuada, dolor enim laoreet lectus, a tincidunt lorem lorem in lorem. Suspendisse sagittis sem ex.

Universidade Federal de Minas Gerais - UFMG

Escola de Engenharia

Programa de Graduação em Engenharia Elétrica

Orientador: Antônio de Pádua Braga

Belo Horizonte, Minas Gerais - Brasil

2018

João Victor Barbosa Alves

Seleção Incremental de Variáveis para Aprendizado de Máquina utilizando Preditor Linear e Validação Cruzada

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus lacinia, erat vitae pulvinar malesuada, dolor enim laoreet lectus, a tincidunt lorem lorem in lorem. Suspendisse sagittis sem ex.

Trabalho aprovado. 1 de novembro de 2018:

Antônio de Pádua Braga
Orientador

Professor
Convidado 1

Professor
Convidado 2

Belo Horizonte, Minas Gerais - Brasil
2018

São as perguntas que não sabemos responder que mais nos ensinam.
Elas nos ensinam a pensar.
Se você dá uma resposta a um homem, tudo o que ele ganha é um fato qualquer.
Mas, se você lhe der uma pergunta, ele procurará suas próprias respostas.
(. . .)
Assim, quando ele encontrar as respostas, elas lhe serão preciosas.
Quanto mais difícil a pergunta, com mais empenho procuramos a resposta.
Quanto mais a procuramos, mais aprendemos.
Rothfuss, Patrick - O Temor do Sábio - a Crônica do Matador do Rei - Segundo Dia.

Resumo

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Palavras-chave: latex. abntex. editoração de texto.

Sumário

1 Introdução

O aumento da capacidade de armazenamento e processamento de dados tem possibilitado o desenvolvimento de sistemas inteligentes através do aprendizado de máquina. Tal desenvolvimento, por sua vez, permitiu avanços em diversas áreas da computação, microeletrônica e sensoriamento.

Hoje, aplicações tais como pesquisas web, sistemas *anti-spam*, reconhecimento de voz, recomendações de produto e diversas outras são frutos desse progresso. Porém a disponibilidade de quantidades massivas de dados, apresenta não só um horizonte vasto de possíveis aplicações, mas também desafios para seleção e processamento desses dados.

1.1 Aprendizado de Máquina

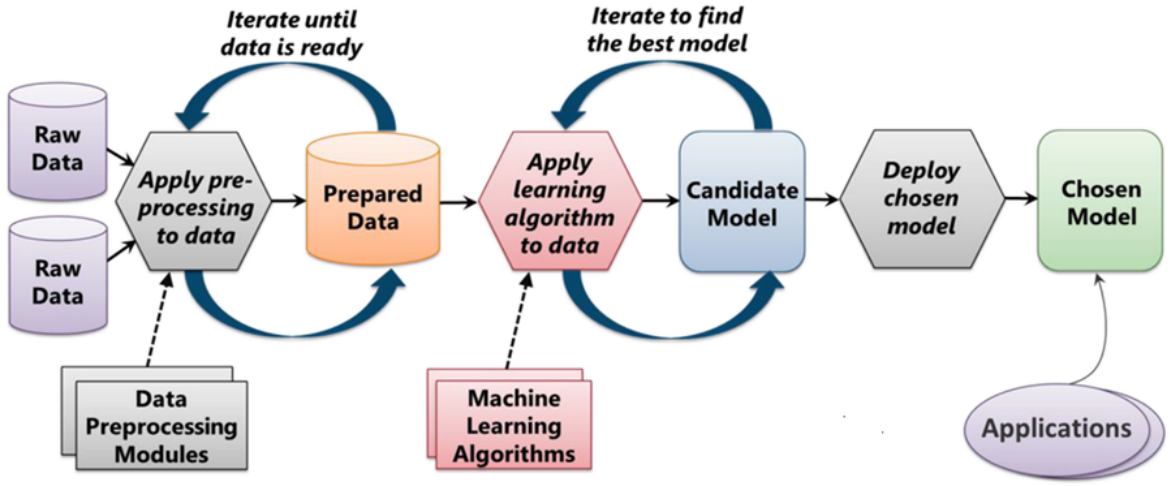
Aprendizado de máquina é o campo da computação responsável por desenvolver e empregar sistemas ou modelos que, através da sua exposição à experiências, são capazes de melhorar sua performance na realização de determinada tarefa (MITCHELL, 1997).

O processo de desenvolvimento de modelos com aprendizado de máquina pode ser dividido, em linhas gerais, em duas grandes etapas.

Inicialmente os dados disponíveis são analisados e tratados. Nessa etapa procura-se identificar correlações entre as informações disponíveis e as variáveis de interesse. Além disso, são explorados possíveis filtros, transformações e outros algoritmos que possam facilitar o aprendizado do modelo. Também é necessário realizar nessa etapa a separação dos dos dados que serão utilizados para treinamento, validação e teste no decorrer do processo.

Na etapa seguinte objetiva-se obter um modelo capaz de reproduzir o comportamento do sistema que gerou o conjunto de dados. Para tal, um ou mais modelos e algoritmos são selecionados e o treinamento é realizado. Duas características são de fundamental importância e devem ser controladas: a capacidade de assimilação do comportamento representado pelos dados (complexidade) e a capacidade de extrapolação das saídas para novas entradas (generalização).

Figura 1 – ADAPTAR: Fluxo de desenvolvimento de sistemas inteligentes.



Fonte: *Introduction to Microsoft Azure by David Chappell.*

Os algoritmos utilizados nesta segunda etapa podem ser classificados em uma de três categorias, de acordo com as características dos dados disponíveis. (RUSSELL; NORVIG, 2010)

Na categoria de aprendizado não supervisionado, desenvolve-se sistemas capazes identificar padrões implícitos em um conjunto de dados não rotulados.

No aprendizado por reforço, os sistemas se adaptam de acordo com dados de resposta oriundos do ambiente no qual estão envolvidos.

Por fim, no aprendizado supervisionado, estão modelos que, a partir de um conjunto de entradas e saídas conhecidas, são capazes de extrapolar a dinâmica do sistema em questão.

1.2 Generalização

Generalização é o termo usado para descrever a capacidade de um sistema de reagir a novos dados. Isto é, após realizado o treinamento, a capacidade de um sistema de realizar predições precisas para dados nunca observados.

O processo de aprendizado supervisionado é a inferência de um mapeamento de dados de entrada a variáveis de saída a partir de um conjunto de observações. Este pode, portanto, ser interpretado como um ajuste não linear de uma curva (HAYKIN, 2009). Consequentemente, os modelos obtidos através desse processo também estão sujeitos a *overfitting* e *underfitting*.

Overfitting, ou sobre-ajuste, é caracterizado pela alta performance no conjunto de dados de treinamento e baixa performance em dados nunca observados. Isto é, o modelo assimila desvios causados por erros de medição ou fatores aleatórios presentes no treinamento. Dessa maneira, o erro em relação aos dados de treinamento é reduzido, porém essa melhora não corresponde a uma melhora na representação da realidade.

De maneira similar, o *underfitting*,ou sub-ajuste, é caracterizado pela baixa performance em ambos os conjuntos de dados, indicando que o modelo utilizado não é capaz de representar de maneira satisfatória a complexidade do sistema real.

Figura 2 – Exemplo de underfitting, ajuste ideal e overfitting.

Fonte:

Enquanto a escolha de um modelo adequado é suficiente para evitar o *underfitting*, diversos fatores contribuem podem contribuir para a ocorrência de *overfitting*.

1.3 Objetivos

Esse trabalho apresenta um método de seleção de variáveis de entrada para modelos treinados através de aprendizado supervisionado.

O algoritmo a ser descrito estima o desempenho das variáveis individualmente e em conjunto através do treinamento de modelos lineares e do cálculo de sua performance aplicando a estratégia de validação cruzada.

2 Revisão de literatura

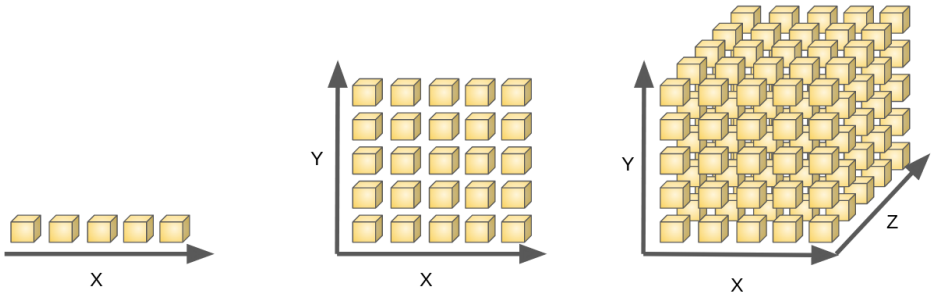
Nesta seção são abordados conceitos fundamentais para a compreensão deste trabalho, assim como tendências atuais da literatura em relação ao tema estudado.

2.1 O problema da dimensionalidade

Sistemas que se utilizam de aprendizado de máquina apresentam respostas baseados nos dados de entrada que lhe são apresentados. Dessa maneira, a determinação de que variáveis são relevantes para o sistema é uma etapa fundamental de seu processo de desenvolvimento.

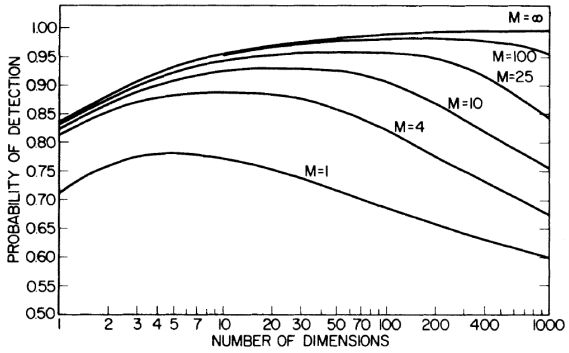
A inclusão de variáveis irrelevantes resulta no aumento da complexidade do modelo, o tornando mais suscetível ao *overfitting* e menos interpretável (JAMES et al., 2017, p. 204). Além disso, um número elevado de variáveis de entrada resulta em um efeito conhecido como a maldição da dimensionalidade (*curse of dimensionality*), onde o aumento do número de dimensões (quantidade de variáveis), implica no aumento exponencial do número de amostras necessárias para representar o espaço de variáveis (BISHOP, 2006, p. 34).

Figura 3 – Crescimento do espaço de variáveis em virtude do aumento das dimensões.



Fonte: The Curse of Dimensionality (GLEESON, 2017)

Figura 4 – Performance em relação ao número de dimenções e amostras.



Fonte: A Problem of Dimensionality: A Simple Example (TRUNK, 1979)

É desejável então que se utilize o menor número possível de variáveis de entrada que permitam ao modelo produzir a resposta correta. Uma maneira de realizar essa redução de dimensionalidade é selecionar um subconjunto das variáveis disponíveis, descartando aquelas que apresentarem pouca ou nenhuma relevância para o problema (JAMES et al., 2017, p. 204).

2.2 Regressão Linear

Os modelos de regressão linear constituem uma classe de modelos que utilizam funções lineares com parâmetros ajustáveis. O exemplo mais simples dessa classe é a função que realiza a combinação linear das entradas com os parâmetros ajustados para gerar uma predição (eq. 2.1).

$$\hat{y} = w_0 + w_1 * x_1 + ... + w_D x_D = \sum_{i=0}^{D} w_i \phi(x_i) = \phi(X)W \tag{2.1}$$

Onde D corresponde ao número de dimensões da variável de entrada, $x_0 = 1$ e $\phi(x_i) = x_i$.

Modelos mais complexos e de maior aplicabilidade podem ser obtidos ao se considerar um conjunto fixo de transformações não-lineares ($\phi_n(x)$) ao invés das variáveis originais, ou em conjunto com elas. Esses modelos são lineares em relação as suas variáveis independentes, porém são não-lineares em relação as variáveis de entrada (BISHOP, 2006).

2.2.1 Treinamento de modelos lineares

O treinamento de modelos lineares consiste em encontrar o vetor de parâmetros W que maximize a similaridade entre o modelo e sistema modelado. Esse treinamento é

normalmente realizado minimizando-se o somatório dos erros quadráticos (eq. 2.2) em função do vetor W.

$$\begin{aligned} E_{sq}(W) &= \frac{1}{2}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2 \\ &= \frac{1}{2}(Y-\hat{Y})^T(Y-\hat{Y}) \\ &= \frac{1}{2}\left(Y^TY - 2\hat{Y}^TY + \hat{Y}^T\hat{Y}\right) \\ &= \frac{1}{2}\left(Y^TY - 2W^T\phi(X)^TY + \sum_{n=1}^{N}(\phi(X_n)W)^2\right) \end{aligned} \tag{2.2}$$

$$\begin{aligned} \frac{\partial^2 E_{sq}(W)}{\partial W \partial} &= \frac{1}{2}\left(0 - 2\phi(X)^TY + 2\sum_{n=1}^{N}\left(\phi(X_n)^T\phi(X_n)\right)W\right) \\ &= -\phi(X)^TY + \phi(X)^T\phi(X)W \end{aligned} \tag{2.3}$$

Igualando-se a eq. 2.3 a zero e isolando o vetor W, obtém-se a eq. 2.4, conhecida como a equação normal para o problema dos mínimos quadrados.

$$\begin{aligned} \phi(X)^T\phi(X)W &= \phi(X)^TY \\ W &= (\phi(X)^T\phi(X))^{-1}\phi(X)^TY \\ W &= \phi(X)^+Y \end{aligned} \tag{2.4}$$

O termo $(\phi(X)^T\phi(X))$ pode se aproximar de uma matriz singular se muitas das variáveis envolvidas forem linearmente dependentes, resultando assim em dificuldades para o cálculo numérico dos valores e possivelmente em um vetor de parâmetros de alta magnitude. Um termo de regularização na eq. 2.2 resolve esse problema, garantindo que a nova matriz não é singular. A dedução da equação normal com termo de regularização é análoga à apresentada e o resulta na eq. 2.5.

$$W = (\lambda I + \phi(X)^T\phi(X))^{-1}\phi(X)^TY = A^{-1}\phi(X)^T_{[i]}Y \tag{2.5}$$

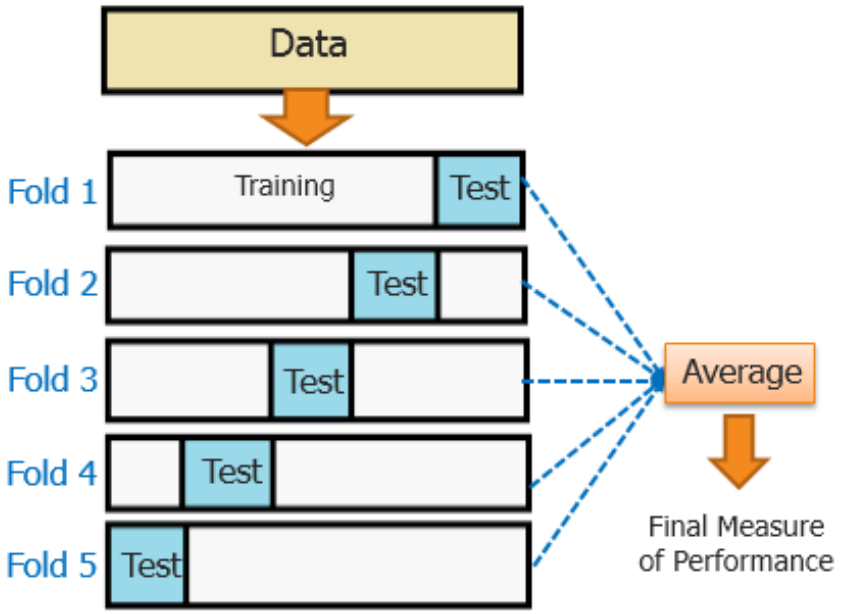
A adição do termo de regularização tem ainda o efeito de limitar a complexidade efetiva do modelo, reduzindo o *overfitting* e possibilitando a utilização de conjuntos de dados menores (BISHOP, 2006).

2.3 Validação Cruzada

A validação cruzada (*cross-validation*) é um conjunto de metodologias de treinamento e validação que, de maneira simples e efetiva, permitem a estimativa do erro de generalização do modelo.

No método *k-fold* de validação cruzada, os dados de treinamento são divididos em k conjuntos aleatórios de tamanhos aproximadamente iguais. Realizada essa divisão, a cada iteração do treinamento, os parâmetros do modelo são determinados utilizando k-1 conjuntos e sua performance é avaliada no conjunto restante. O processo se repete até que todos os k conjuntos tenham sido utilizados para avaliação e a média das performances observadas fornecem uma estimativa da performance de generalização do modelo (CAWLEY; TALBOT, 2010).

Figura 5 – Validação cruzada: *5-fold*.



Fonte: Different types of Validations in Machine Learning (Cross Validation) (SRINIDHI, 2018)

Um caso especial do método descrito é o *leave-one-out cross-validation* (LOOCV), onde k é igual ao total de amostras disponíveis (N), ou seja, o treinamento é realizado N vezes com N-1 amostras, e validado na amostra excluída. Tal abordagem apesar de apresentar, em geral, um custo computacional elevado, resulta em um modelo com menor variabilidade e viés para modelos de regressão linear (BURMAN, 1989), produzindo assim resultados com melhor performance e menos *overfitting*.

2.3.1 *Leave-one-out Cross-validation* e Regressão Linear

Especificamente para o caso do preditor linear, é possível determinar o resultado do LOOCV sem a necessidade de se calcular N preditores, tornando essa uma estratégia interessante para determinar a capacidade de generalização de um modelo linear. A seguir uma dedução adaptada de (SEBER; LEE, 2012, p. 268) é apresentada.

O erro de LOOCV corresponde a média dos erros quadráticos de cada estimador obtido excluindo-se uma amostra.

$$LOOCV = \frac{1}{N}\sum_{i=1}^{N} e_{[i]}^2 \tag{2.6}$$

$$e_{[i]} = y_i - \hat{y}_{[i]} = y_i - x_i W_{[i]} \tag{2.7}$$

Onde o i indexa a amostra excluída no treinamento, $\hat{y}_{[i]}$ corresponde à saída do preditor calculado sem a amostra (x_i, y_i) e $W_{[i]}$ os parâmetros desse preditor.

Pode-se definir o vetor de parâmetros $W_{[i]}$ conforme a eq. 2.8.

$$W_{[i]} = A_{[i]}^{-1}\phi(X)_{[i]}^T Y_{[i]} \tag{2.8}$$

É possivel perceber as seguintes igualdades:

$$\begin{aligned} A_{[i]} &= \lambda I + \phi(X)_{[i]}^T \phi(X)_{[i]} \\ &= \lambda I + \phi(X)^T\phi(X) - x_i^T x_i \\ &= A - x_i^T x_i \end{aligned} \tag{2.9}$$

$$\phi(X)_{[i]}^T Y_{[i]} = \phi(X)^T Y - x_i^T y_i \tag{2.10}$$

Aplicando a formula de Sherman–Morrison à eq. 2.9, temos:

$$\begin{aligned} h_i &= x_i A^{-1} x_i^T \\ A_{[i]}^{-1} &= A^{-1} + \frac{A^{-1}x_i x_i^T A^{-1}}{1 - h_i} \end{aligned} \tag{2.11}$$

Substituindo as eq. 2.10 e 2.11 na eq. 2.8.

$$\begin{aligned} W_{[i]} &= \left(A^{-1} + \frac{A^{-1}x_i^T x_i A^{-1}}{1-h_i}\right)\left(\phi(X)^T Y - x_i^T y_i\right) \\ &= W - A^{-1}x_i^T y_i + \frac{A^{-1}x_i^T x_i W}{1-h_i} - \frac{A^{-1}x_i^T x_i A^{-1} x_i^T y_i}{1-h_i} \\ &= W - \frac{A^{-1}x_i^T}{1-h_i}\left(y_i(1-h_i) - x_i W + h_i y_i\right) \\ &= W - \frac{A^{-1}x_i^T}{1-h_i}\left(y_i - \hat{y}_i\right) \end{aligned} \tag{2.12}$$

Substituindo as eq. 2.12 na eq. 2.7.

$$\begin{aligned} e_{[i]} &= y_i - x_i W_{[i]} \\ &= y_i - x_i \left(W - \frac{A^{-1} x_i^T}{1 - h_i} (y_i - \hat{y}_i) \right) \\ &= y_i - \hat{y}_i + \frac{h_i}{1 - h_i} (y_i - \hat{y}_i) \\ &= \frac{y_i - \hat{y}_i}{1 - h_i} \end{aligned} \tag{2.13}$$

Seja P a matriz de projeção (BASILEVSKY, 2005, p. 303) do preditor (também conhecida como matriz chapéu (SEBER; LEE, 2012, p. 266)) e a matriz R definida conforme a eq. 2.14

$$\begin{aligned} \hat{Y} = PY \implies P = XA^{-1}X^T \\ RY = Y - \hat{Y} \implies R = I - P = I - XA^{-1}X^T \end{aligned} \tag{2.14}$$

Para cada índice i, o numerador da eq. 2.13 corresponde a linha de mesmo índice da matriz RY. Além disso o denominador da equação corresponde ao elemento da diagonal da matriz R de mesmo índice. A partir dessas observações, pode-se agrupar todos os valores de $e_{[i]}$ na forma da matriz E_{LOOCV}, conforme a equação 2.15.

$$E_{LOOCV} = diag(R)^{-1} RY \tag{2.15}$$

Dessa forma a eq. 2.6 pode ser reescrita conforme eq. 2.16.

$$\begin{aligned} LOOCV &= \frac{1}{N} \sum_{i=1}^{N} e_{[i]}^2 \\ &= \frac{1}{N} E_{LOOCV}^2 \\ &= \frac{1}{N} E_{LOOCV}^T E_{LOOCV} \\ &= \frac{1}{N} Y^T R \; diag(R)^{-2} RY \end{aligned} \tag{2.16}$$

3 Materiais e métodos

4 Resultados

5 Conclusão

Referências

BASILEVSKY, A. *Applied matrix algebra in the statistical sciences*. [S.l.]: Dover Publications, 2005. Citado na página 14.

BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. Citado 3 vezes nas páginas 9, 10 e 11.

BURMAN, P. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, v. 76, n. 3, 1989. Citado na página 12.

CAWLEY, G. C.; TALBOT, N. L. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.*, JMLR.org, v. 11, p. 2079–2107, ago. 2010. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1756006.1859921>. Citado na página 12.

GLEESON, P. *The Curse of Dimensionality*. Medium, 2017. Disponível em: <https://medium.freecodecamp.org/the-curse-of-dimensionality-how-we-can-save-big-data-from-itself-d9fa0f872335>. Acesso em: 31/10/2018. Citado na página 9.

HAYKIN, S. S. *Neural networks and learning machines*. 3. ed. [S.l.]: Prentice Hall, 2009. Citado na página 7.

JAMES, G. et al. *An introduction to statistical learning with applications in R*. [S.l.]: Springer, 2017. Citado 2 vezes nas páginas 9 e 10.

MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-Hill, 1997. Citado na página 6.

RUSSELL, S. J.; NORVIG, P. *Artificial intelligence - A Modern Approach*. 3. ed. [S.l.]: Prentice Hall, 2010. Citado na página 7.

SEBER, G. A. F.; LEE, A. J. *Linear regression analysis*. 2. ed. [S.l.]: Wiley, 2012. Citado 2 vezes nas páginas 13 e 14.

SRINIDHI, S. *Different types of Validations in Machine Learning (Cross Validation)*. 2018. Disponível em: <https://blog.contactsunny.com/data-science/different-types-of-validations-in-machine-learning-cross-validation>. Acesso em: 31/10/2018. Citado na página 12.

TRUNK, G. V. A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1, n. 3, p. 306–307, July 1979. ISSN 0162-8828. Citado na página 10.