UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

JEAN CARLLO JARDIM COSTA

UMA ANÁLISE SOBRE O IMPACTO DE DADOS FALTANTES NO DESEMPENHO DE MÉTODOS DE APRENDIZADO DE MÁQUINA

FORTALEZA

2018

JEAN CARLLO JARDIM COSTA

UMA ANÁLISE SOBRE O IMPACTO DE DADOS FALTANTES NO DESEMPENHO DE MÉTODOS DE APRENDIZADO DE MÁQUINA

Dissertação apresentada ao Curso de do Programa de Pós-Graduação em Ciências da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. João Paulo Pordeus Gomes

FORTALEZA

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C873a Costa, Jean Carllo Jardim.
Uma análise sobre o impacto de Dados Faltantes no desempenho de métodos de Aprendizado de Máquina / Jean Carllo Jardim Costa. – 2018.
55 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2018.
Orientação: Prof. Dr. João Paulo Pordeus Gomes.

1. Dados Faltantes. 2. Aprendizado de Máquina. 3. Meta-Aprendizado. I. Título.

CDD 005

JEAN CARLLO JARDIM COSTA

UMA ANÁLISE SOBRE O IMPACTO DE DADOS FALTANTES NO DESEMPENHO DE MÉTODOS DE APRENDIZADO DE MÁQUINA

Dissertação apresentada ao Curso de do Programa de Pós-Graduação em Ciências da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 27 de novembro de 2018

BANCA EXAMINADORA

Prof. Dr. João Paulo Pordeus Gomes (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Maria da Silva Monteiro Filho
Universidade Federal do Ceará (UFC)

Prof. Dr. Leonardo Ramos Rodrigues
Instituto de Aeronáutica e Espaço (IAE)

Dedico essa dissertação à minha mãe, Mara, ao meu pai, Gabriel, e a minha esposa, Danielly e ao meu filho Bento.

AGRADECIMENTOS

Ao Prof. Dr. João Paulo Pordeus Gomes por me orientar em minha dissertação e estar disponível sempre que necessário.

À minha mãe por sempre acreditar e me manter firme nos estudos. Ao meu Pai, pelas palavras de incentivo.

À minha esposa que não mediu esforços parta estar sempre ao meu lado me apoiando com o que fosse necessário para conclusão de mais esta etapa na minha vida.

Aos meus familiares, que mesmo distantes torcem pelo meu sucesso.

Aos colegas do LIA que sempre proporcionaram boas discussões acadêmicas contribuindo com sugestões e ideias para a dissertação.

Aos meus amigos que sempre me incentivaram e faziam com que os momentos de tanta exigência do ambiente acadêmico parecessem mais simples e felizes.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

Ao Departamento de Computação, ao MDCC e aos seus professores e colaboradores pelo suporte técnico e científico que me propiciaram alcançar mais essa realização na minha vida.

A todos que me ajudaram, direta ou indiretamente, o meu mais sincero agradecimento.

"A menos que modifiquemos a nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo."

(Albert Einstein)

RESUMO

A ocorrência de dados faltantes é um problema recorrente e tem despertado interesse de pesquisadores ao longo das últimas décadas. Devido a isto, muitos métodos para imputação de dados têm sido propostos. Nesta dissertação é apresentado um estudo do impacto da aplicação de vários métodos de imputação de dados faltantes no desempenho de métodos de aprendizado de máquina, tanto classificação como regressão. O resultado obtido mostra que os algoritmos de imputação podem ter impacto relevante no desempenho de algoritmos de classificação e regressão dependendo do percentual de dados faltantes. Adicionalmente é apresentado um modelo para recomendação de algoritmos de imputação de dados que compara três classificadores (Florestas Aleatórias, Gradiente Boosting e Máquina de Vetor de Suporte) no desenvolvimento desta tarefa onde ambos apresentam bons resultados.

Palavras-chave: Dados Faltantes. Aprendizado de Máquina. Meta-Aprendizado.

ABSTRACT

The occurrence of missing data is a recurrent problem and it has aroused the interest of researches over the last decades. Hence, many imputation methods have been proposed in recent years. In this dissertation, we present a study about the impact of the application of several imputation methods on the performance of machine learning algorithms, for both classification and regression. The result obtained shows that the imputation algorithms can have a relevant impact on the performance of classification and regression algorithms depending on the percentage of missing data. In addition, a model for the recommendation of data imputation algorithms is presented, which compares three classifiers (Random Forests, Gradient Boosting and Support Vector Machine) in the development of this task where both have good results.

Keywords: Missing Data. Machine Learning. Meta-Learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CD	*Critical Difference*
IA	*Iterative Algorithm*
ICKNNI	*Incomplete-Case Nearest Neighbor Imputation*
MAR	*Missing at Random*
MCAR	*Missing Completely at Random*
MLP	*Multilayer Perceptron*
NIPALS	*Nonlinear Iterative Partial Least Squares Regression Algorithm*
NMAR	*Not Missing at Random*
PCA	*Principal Components Analisys*
SVM	*Support Vector Machine*

LISTA DE SÍMBOLOS

X	Base de dados.
x_i	Vetor com os elemento da linha i da base de dados X.
X_{ij}	i-ésimo elemento da j-ésima coluna da base de dados X.
V	Vetor de rótulos.
v_i	Elemento i do vetor de rótulos V
w	Vetor de pesos.
W	Matriz de pesos.
n	Número de instancias.
ξ_i	Variável de folga da instância i.
C	Parâmetro de regularização.
$\mathbb{R}^d$	Espaço dos números Reais com d dimensões.
k	Constante.
$y_m(X)$	m-ésimo classificador aplicado sobre a base de dados X.
$F(x)$	Função aproximada.
$F^*(x)$	Função real.
min	Mínimo.
g	Função de gradiente.
L	Função de perda.
$h_m(x)$	m-ésima função de perda.
φ	Função de ativação.
y_k	k-ésimo perceptron.
M	Matriz indicadora de dados faltantes.
m_t	t-ésimo vetor de médias da base de dados X.
m_t^T	t-ésimo vetor de médias da base de dados X transposto.
M_{ij}	i-ésimo elemento da j-ésima coluna da matriz indicadora de dados faltantes.
Z	Matriz de dados.

Z_{obs}	Valores observados de Z.
Z_{mis}	Valores não observados de Z.
H_0	Hipótese nula.
P	Probabilidade.
R	Matriz estimada após aplicação do PCA.
$R_{(mis)i}$	Vetor dos valores perdidos da linha i da matriz R.
$R_{(obs)i}$	Vetor dos valores observados da linha i da matriz R.
S_{ij}	Conjunto das k instância mais próximas da instância Z_i com valor observável no atributo j.

SUMÁRIO

1 INTRODUÇÃO

O interesse em aprendizado de máquina tem crescido recentemente devido ao seu grande sucesso em aplicações reais dos mais diversos tipos. Fundamentalmente podemos defini-lo como um conjunto de algoritmos usados para extrair informações de dados brutos e representá-los em algum tipo de modelo, este, por sua vez, é usado para inferir algo sobre dados que ainda não foram modelados(PATTERSON; GIBSON, 2017). Essa técnica pode ser utilizada em uma infinidade de cenários, tais como softwares médicos no auxílio ao diagnóstico, classificação de risco de empréstimo por instituições bancárias, recomendação de produtos e serviços, detecção de spams.

Um problema muito recorrente para quem lida com aprendizagem de máquina é a ocorrência de dados faltantes. Os valores podem estar ausentes na base de dados por diversos fatores, incluindo erro de medição, mal funcionamento de dispositivos, falhas de operação e muitos outros (SOVILJ *et al.*, 2016).

A forma mais fácil de tratar o problema de dados faltantes é descartar todas as entradas que apresentam algum atributo faltando. No entanto, simplesmente apagar os dados pode gerar um impacto muito negativo no resultado, conduzindo a predições erradas. Além disso, se a quantidade de dados faltantes for muito grande, pode ser impossível a aplicação de qualquer método de aprendizagem de máquina. Outra estratégia, mais adequada na maioria dos casos, é preencher os dados faltantes (imputar) seguindo alguma técnica comprovadamente eficiente para, desse modo, aproveitar toda informação disponível, melhorando as predições.

Ao longo dos anos, muitos métodos de imputação foram propostos, como por exemplo, preencher com a média dos dados, *Incomplete-Case Nearest Neighbor Imputation* (ICKNNI) (HULSE; KHOSHGOFTAAR, 2011), *Iterative Algorithm* (IA), Modified *Nonlinear Iterative Partial Least Squares Regression Algorithm* (NIPALS), (FOLCH-FORTUNY *et al.*, 2016), entre outros. Estes métodos apresentam resultados promissores, porém não é possível verificar que qualquer desses métodos apresenta melhores resultados em todas as aplicações, sendo cada um deles melhores com determinadas bases de dados e piores com outras.

Outro problema muito comum para quem lida com dados faltantes é escolher dentre os métodos de imputação aquele que obtém o melhor resultado em uma determinada base dados. Uma solução seria testar todos os métodos e então escolher o com melhor desempenho, no entanto a tarefa executada dessa forma pode conduzir a um trabalho demasiadamente dispendioso. Uma solução mais adequada é através de um sistema de recomendação baseado em meta-aprendizado,

no qual o sistema, baseado em meta-atributos diz qual o possível melhor algoritmo para dada base de dados.

A contribuição deste trabalho é fazer a comparação do impacto de métodos de imputação no desempenho de métodos de aprendizagem de máquina diante de cenários distintos de percentual de dados faltantes em tarefas de regressão e classificação. Adicionalmente é apresentado uma abordagem de meta-aprendizado com o intuito de desenvolver um modelo de recomendação de algoritmos para predizer o método de imputação com melhor desempenho em uma determinada base de dados não utilizada na etapa de aprendizagem.

1.1 Objetivo Geral

O objetivo geral desse estudo é avaliar o impacto de dados faltantes no desempenho de algoritmos de classificação e regressão utilizando-se para isso vários métodos de imputação de dados. Em aditivo objetiva-se desenvolver um modelo para recomendação de algoritmos de imputação de dados.

1.2 Objetivos Específicos

Utilizar variados métodos de imputação de dados em bases com dados faltantes gerados artificialmente para em seguida aplicar algoritmos de aprendizagem de máquina em problemas de classificação e regressão. Na sequência deverá ser produzida uma análise do desempenho destes algoritmos.

Por outro lado objetiva-se criar uma base de dados contendo os meta-atributos referentes a problemas de classificação e regressão com o respectivo algoritmo com melhor desempenho na tarefa de imputação. Por fim um classificador deve ser usado para recomendar o melhor algoritmo de imputação dado uma base nova e seus respectivos meta-dados.

1.3 Contribuições do Autor

Este trabalho desenvolve uma comparação de métodos de imputação diante de vários cenários, avaliando seus impactos em métodos de classificação e regressão. Em um segundo momento é apresentado um modelo que realiza a extração meta-dados de várias bases, apresenta os algoritmos de imputação com melhor desempenho nas respectivas bases e por fim compara o desempenho de dois classificadores na tarefa de recomendação de algoritmo de imputação.

1.4 Organização da Dissertação

Esta dissertação está organizada da seguinte forma: N capítulo 2 são apresentados os fundamentos teóricos referentes a aprendizagem de máquina com foco voltado para modelos de classificação e regressão de dados utilizados. No capítulo 3 é feito um desenvolvimento teórico sobre dados faltantes, apresentando os mecanismos geradores, além de algoritmos de imputação referentes a este trabalho. O capítulo 4 trata de meta-aprendizado e os modos de se obter meta-atributos. O capítulo 5 apresenta a metodologia desenvolvida para obtenção dos resultados. Os resultados obtidos são discutidos no capítulo 6. Por fim, o capítulo 7 traz as considerações finais e os trabalhos futuros.

2 APRENDIZAGEM DE MÁQUINA

Neste capítulo abordaremos os principais aspectos referentes à aprendizagem de máquina, partindo de uma abordagem mais superficial para apresentar os principais conceitos, passando pelos principais subcampos e focando principalmente nos métodos de classificação (*Support Vector Machine* (SVM), os Métodos de Comitê Florestas Aleatórias e Gradiente Boosting) e regressão (*Multilayer Perceptron* (MLP)), pois estes métodos citados foram utilizados neste trabalho.

Os capítulos subsequentes estão dispostas da seguinte forma: No capítulo 2.1 será abordado alguns fundamentos de aprendizagem de máquina; no capítulo 2.2 será abordado classificação e os respectivos algoritmos de classificação utilizados neste trabalho; No capítulo 2.3 a abordagem é voltada para regressão e o algoritmo utilizado neste trabalho.

2.1 Fundamentos

Quando se fala em aprendizagem de máquina nós temos basicamente três tipos principais de abordagem: aprendizagem supervisionada, que será discutido em detalhes nas subseções seguintes, pois é este tipo de aprendizagem que sustenta este trabalho; aprendizagem não supervisionada; aprendizagem por reforço.

O objetivo principal da aprendizagem supervisionada é aprender um modelo a partir de dados de treinamento rotulados que nos permitirão fazer predições sobre dados não vistos ou futuros. O termo supervisionado refere-se a um conjunto de exemplos onde o rótulo já é conhecido (RASCHKA, 2015). Como exemplo de aplicação desta abordagem podemos citar a classificação de opiniões sobre um determinado filme, ou seja, primeiro rotulamos algumas opiniões em positivas ou negativas e depois um algoritmo de aprendizagem utiliza essas informações para treinar e posteriormente dizer se novas opiniões, não utilizadas no treinamento, são positivas ou negativas.

Na tarefa de aprendizagem não supervisionada, por sua vez, a extração do conhecimento se dá a partir de exemplos não rotulados. Para (HARRINGTON, 2012) podemos destacar três tipos principais dessa abordagem: agrupamento, onde separamos em grupos exemplares similares; estimativa de densidade, que se refere aos valores estatísticos que descreve os dados; A outra tarefa é reduzir as dimensões de muitos atributos para poucos.

Para entender a aprendizagem por reforço imagine um jogo de xadrez, onde um

agente precisa saber se cada jogada é boa ou ruim, no entanto não existe nenhuma instrução nesse sentido. Logo, sem nenhum feedback, o agente precisa precisa descobrir com seu próprio conhecimento quando acidentalmente faz uma jogada boa (xeque-mate), ou jogada ruim (quando perde o jogo). Este tipo de retorno é chamado de recompensa ou reforço, ou seja, a tarefa de aprendizagem por reforço é usar recompensas observadas para aprender uma política ótima (ou quase ideal) para ambiente (RUSSELL; NORVIG, 2012).

2.2 Classificação

Para (RASCHKA, 2015) classificação é uma subcategoria de aprendizagem supervisionada no qual o objetivo é predizer o rótulo de uma classe categórica de novas instâncias baseados em observações passadas. Esses rótulos de classes são discretos, com valores desordenados que pode ser entendido como os membros de grupo de instâncias.

Uma classificação muito comum de variáveis é em quantitativa ou qualitativa, esta última também conhecida como categórica. Alguns exemplos de dados qualitativos são: gênero (masculino ou feminino); classes sociais (A, B, C, D e E); marcas de carros postos a venda; tipos de câncer diagnosticados (mama, próstata, pulmão). Desse modo, uma associação muito comum é entre tarefa de classificação e variáveis qualitativas, embora isto nem sempre seja tão nítido (JAMES *et al.*, 2013).

2.2.1 Support Vector Machine

Suponha que temos n pontos de dados X_i em um espaço $\mathbb{R}^d$ com seus respectivos rótulos (labels) v_i com $i \in [1..n]$ representando a classe de cada elemento, então podemos aplicar uma das técnicas mais conhecidas, basilares e muito útil para classificação de dados que é a SVM. Ela constitui um método supervisionado, no qual tenta-se separar duas classes através de um hiperplano. Desse modo os dados que ficam de um lado deste hiperplano pertencem a uma classe e os dados do lado oposto pertencem a outra classe.

Uma importante consideração a se fazer é que o hiperplano escolhido detre os mais diversos disponíveis é aquele que separa os pontos das classes com a melhor margem possível como é demonstrado na Figura 1. Para (SUYKENS *et al.*, 2014) margem é a menor distância para o hiperplano entre todos os pontos de dados, ou seja, maximizando esta margem sobre todos os possíveis classificadores lineares w.

Para a aplicação prática bem sucedida de SVM, o conceito de soft-margin para tolerância pontos isolados (*outliers*) é de importância central. Desse modo podemos definir o seguinte problema de otimização:

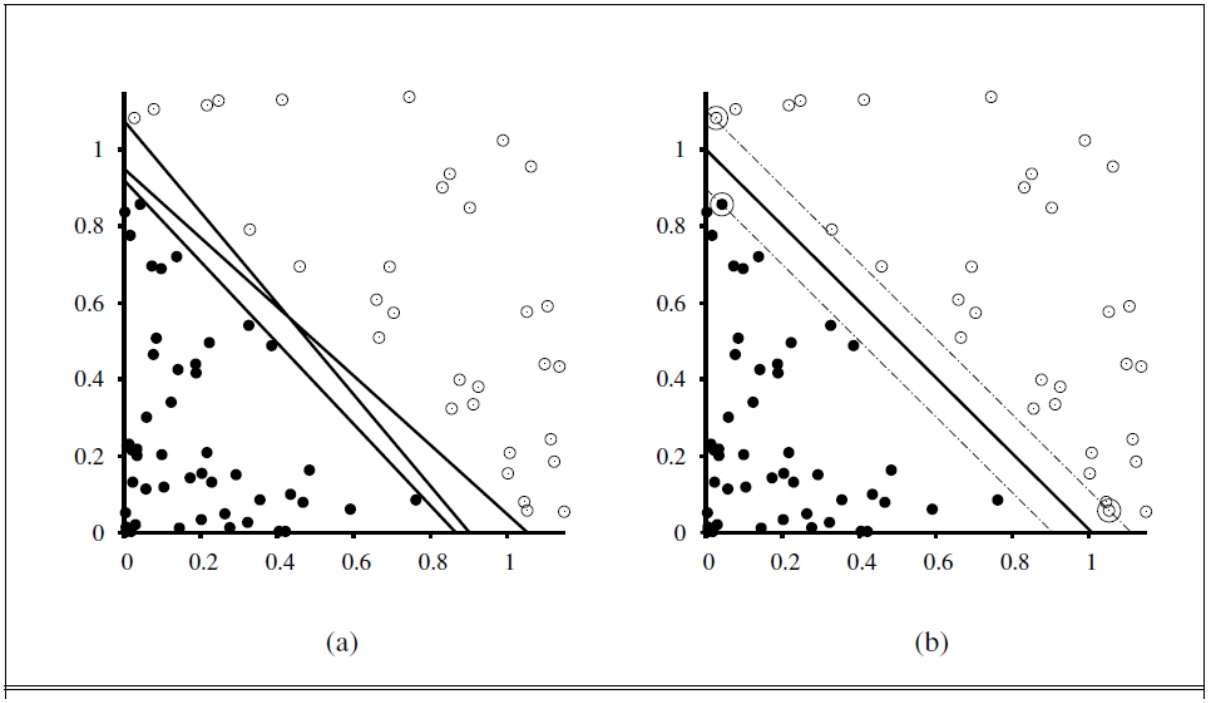
$$\min_{w\in\mathbb{R}^d,\xi in\mathbb{R}^n} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i^2$$

$$s.t.$$

$$v_i(w^T X_i + b) \geq 1 - \xi_i,$$

onde para cada ponto introduzimos uma variável de folga ξ_i, além de uma além de um parâmetro de regularização $C \geq 0$.

Figura 1 – *Support Vector Machine* para classificação: (a) Duas classes (círculos brancos e pretos) e três candidatos a separadores de classes. (b) Separador com margem máxima (linha mais escura), está bem no centro das margens das classes. Os vetores de suporte (pontos com círculos maiores) são os exemplares mais próximos do separador.

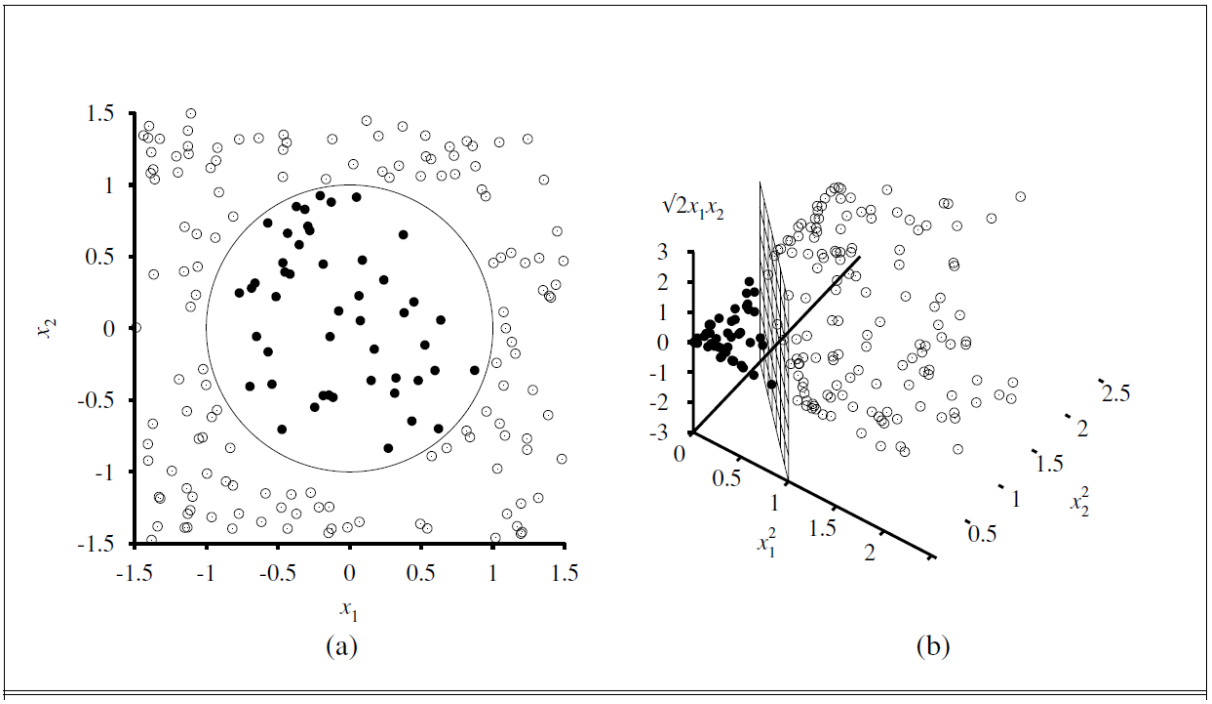


Fonte: (RUSSELL; NORVIG, 2012).

O SVM foi desenvolvido originalmente para lidar com apenas duas classes, no entanto podemos perceber que a grande maioria dos problemas lidam com múltiplas classes. Neste caso a solução encontrada foi transformar o problema multiclasse que se está tratando em vários problemas binários.

Um outro problema enfrentado pelo SVM é o fato de um grande volume de problemas não serem linearmente separáveis, então se aplicarmos o algoritmo em uma base de dados deste tipo não teríamos um resultado satisfatório. Para contornar esta situação, a solução encontrada foi utilizar a kernelização para transformar o espaço não linearmente separável em um espaço de dimensão maior onde seja possível separar por um hiperplano como demonstra a Figura 2 a título de exemplo.

Figura 2 – (a) Um conjunto bidimensional não separável linearmente com o limiar de decisão dado por $x_1^2 + x_2^2 \leq 1$. (b) Mapeamento do espaço bidimensional em um espaço tridimensional $(x_1^2, x_2^2, \sqrt{2x_1x_2})$. A separação circular em (a) se torna uma separação linear em (b)



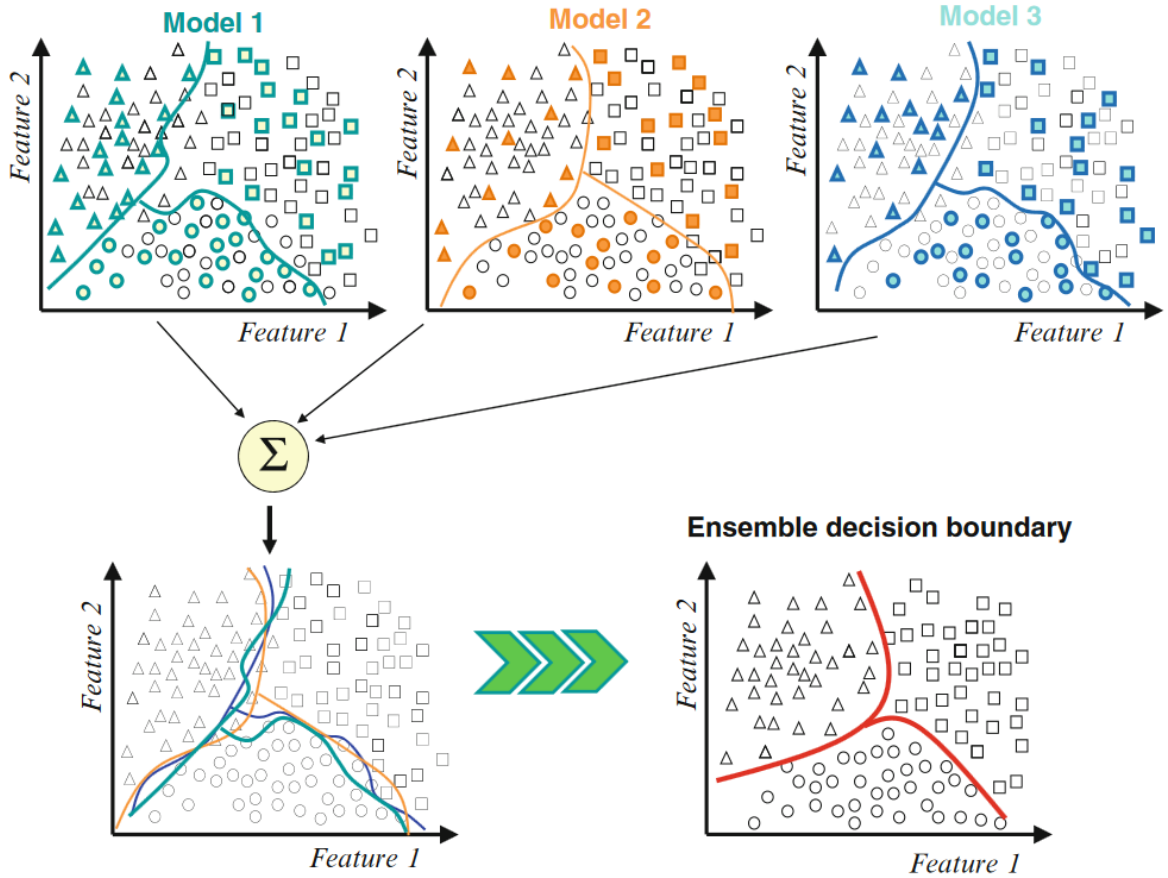
Fonte: (RUSSELL; NORVIG, 2012).

2.2.2 Métodos de Comitê

Nas ultimas décadas, os comitês de classificadores despertaram a atenção de muitos pesquisadores e desenvolvedores, experimentando posição de destaque na área de inteligência computacional e aprendizado de máquina. Apesar do surgimento desses métodos ter sido para resolver o problema da variância, eles têm se mostrado muito efetivos em variados domínios de tarefas e aplicações do mundo real, especialmente nos sistemas de suporte à decisão (ZHANG, 2012). Decisões de comitês pode até parecer novidade, mas na verdade isso já é usado pela humanidade há muitos séculos quando, por exemplo, escolhemos representantes, o Congresso

aprova uma lei. Quando se trata de algoritmos funciona da mesma maneira, ou seja, se temos um problema de decisão, como dizer a qual classe um indivíduo pertence, então é considerado a escolha de vários algoritmos (os melhores) segundo um critério. A Figura 3 demonstra o processo de combinação de classificadores na criação de um modelo único de decisão, no qual três classificadores são treinados e por fim a decisão dos mesmos é combinado para formar um quarto mais eficiente.

Figura 3 – Combinação de modelos de classificação para formar um único modelo



Fonte: (ZHANG, 2012).

Vale ressaltar que se temos um modelo que sempre acerta na decisão, então não precisamos de um comitê para decidir, no entanto é pouco provável que isso ocorra, logo é natural que a decisão de uma maioria de algoritmos seja mais eficiente. Para (ZHANG, 2012) devemos considerar que um modelo de comitê possui três pilares fundamentais: diversidade; treinamento dos membros do conjunto; combinação dos membros.

É possível notar observando a Figura 3 que cada modelo individualmente possui um baixo bias e consequentemente gera uma alta variância, no entanto quando são combinados para gerar um novo modelo a variância diminui, proporcionando uma acurácia maior no grupo de

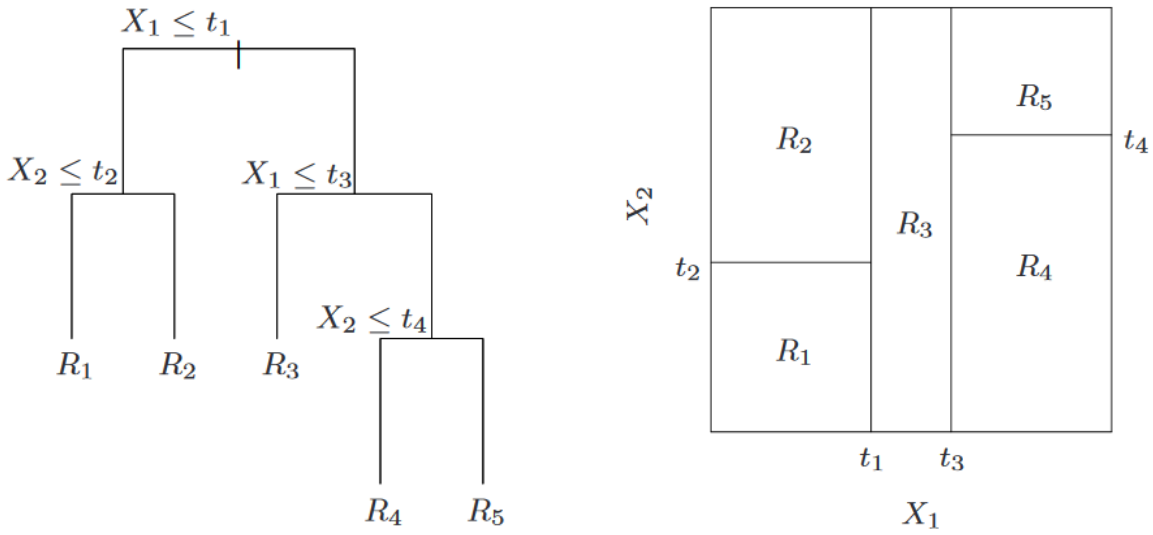
teste.

2.2.2.1 Florestas Aleatórias

Florestas aleatórias se tornou um método de aprendizagem de máquina muito popular devido principalmente ao seu bom desempenho, simplicidade de implementação, capacidade de lidar com alta dimensionalidade, entre outros. Este algoritmo, bem como os métodos de conjunto em geral, são baseados em algoritmos de aprendizagem fracos, porém diversificados, neste caso em particular utiliza-se árvores de decisão. Portanto antes de prosseguir segue um breve introdução deste algoritmo.

Uma maneira muito simples de construir árvores de decisão é de modo *top-down* e de maneira gulosa dividindo desde a raiz, passando pelos nós intermediários até chegar nas folhas, onde uma classe é definida, ou seja, um dado ao passar por um nó é avaliado por uma função de decisão, também chamada de função objetivo, que encaminha recursivamente para um dos nós filhos para novamente ser avaliado até que finalmente chegue a um nó folha que contém a classe a qual aquele dado pertence. A figura 4 demonstra uma árvore de decisão fictícia com os elementos descritos neste parágrafo.

Figura 4 – Em (a) temos uma árvore de decisão binária recursiva com 5 classes distintas (R1-5) com dois atributos (X1, X2) e com a divisão feita nos valores (t1-4). Em (b) é mostrado o espaço de decisão correspondente a árvore.



(a) (b)

Fonte: (TREVOR *et al.*, 2009).

Voltando para o tema principal deste tópico, podemos definir 4 passos básicos para

construção do algoritmo de florestas aleatórias de acordo com (RASCHKA, 2015), são eles:

1. Construa uma amostra aleatória de tamanho *n* usando bootstrap (esta técnica escolhe uma amostra aleatória dos dados do conjunto de treinamento com reposição dos mesmos)
2. Construa uma árvore de decisão a partir dos dados selecinonados na etapa 1. E em cada nó faça o seguinte:
    a) Escolha aleatoriamente *d* atributos sem reposição.
    b) Divida o nó usando o atributo que forneça a melhor divisão de acordo com a função objetivo, por exemplo, maximizando o ganho de informação.
3. Repita os passos 1 e 2 *k* vezes.
4. Agregue a predição de cada uma das *k* árvores criadas anteriormente para decidir a classe pelo voto da maioria das árvores.

Embora florestas aleatórias percam um pouco da interpretabilidade, típicas das árvores de decisão, por outro lado oferecem uma grande vantagem que é não se preocupar tanto em descobrir os melhores hiperparâmetros, bastando principalmente decidir o valor de *k*, pois de modo geral quanto maior for esse valor maior tende ser a acurácia deste método. No entanto vale ressaltar que os recursos computacionais não são infinitos e portanto quanto maior o número de árvores, maior será o tempo de resposta e treinamento do modelo.

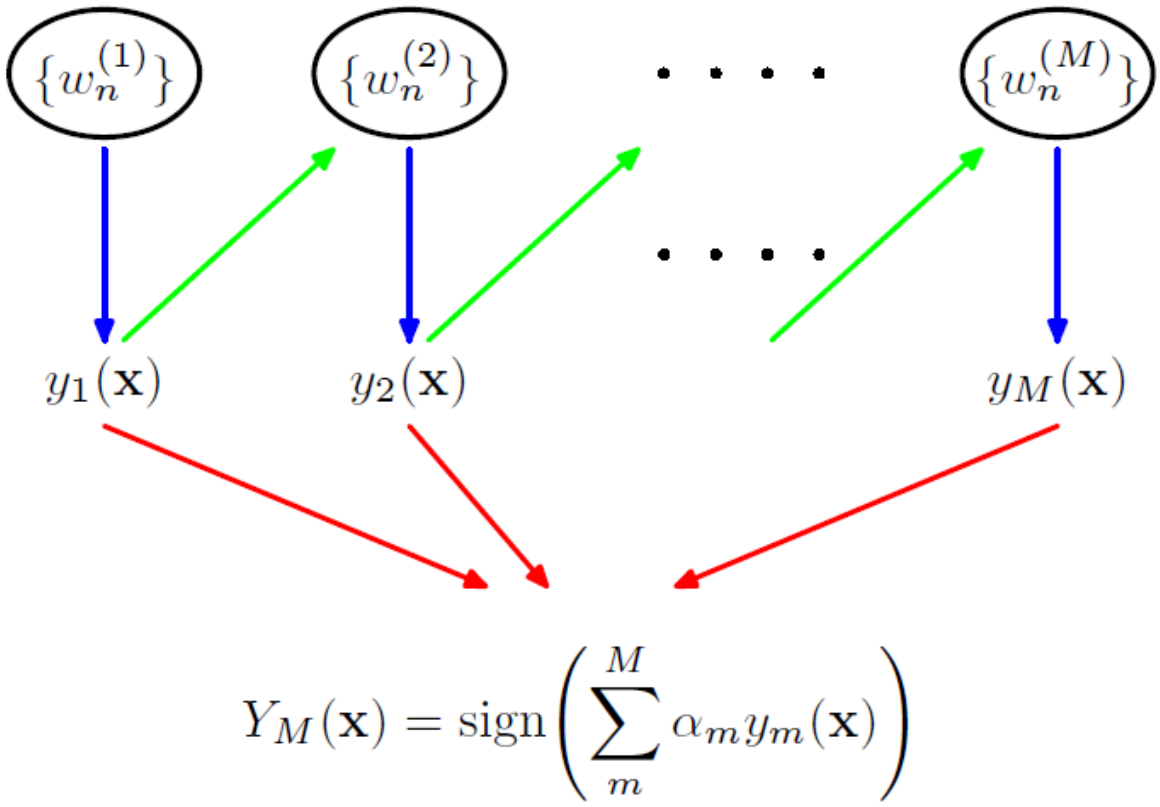
2.2.2.2 Gradiente Boosting

Boosting é uma ideia muito eficiente que foi introduzida em aprendizagem de máquina e que atualmente possui diversas ramificações e implementações distintas sobre a mesma base, sendo Gradiente Boosting uma das mais populares dos últimos anos. Para (BISHOP, 2006) Boosting é uma poderosa técnica para combinar múltiplos classificadores bases para produzir uma forma de comitê cujo desempenho pode ser significativamente melhor do que o de qualquer dos classificadores de base.

Os métodos de boosting funcionam da seguinte forma: vários métodos (classificadores base) são treinado em sequência, onde cada um recebe um coeficiente de ponderação que vai depender do desempenho do classificador imediatamente anterior. Ao final do treinamento os modelos são combinados com um peso associado a cada um, então é criado um esquema de votação, no qual a classe predita é aquela cuja maioria ponderada escolhe, como é mostrado na Figura 5.

Posto isso, podemos adentrar no modelo de classificação Gradiente Boosting. Ima-

Figura 5 – Ilustração esquemática da estrutura do boosting. Cada classificador base $y_m(x)$ é treinado de forma ponderada na base de treinamento (setas azuis) no qual os pesos w_n^m dependem do desempenho dos classificadores bases anteriores $y_{m-1}(x)$ (setas verdes). depois de treinados os classificadores base eles são combinados para dar o classificador final $Y_M(x)$ (setas vermelhas).



Fonte: (BISHOP, 2006).

gine um conjunto de dados $X\{x_1, x_2, ..., x_n\}$ e um conjunto de classes $Y\{y_1, y_2, ..., y_n\}$ correspondente as entradas de X, então este método objetiva descobrir uma função $y = F(x)$ que aproxime a função real $y = F^*(x)$ de modo que a função de perda L seja mínima.

Para se obter a função $F = \sum_{i=1}^{M} h_m$ de gradiente boosting, no qual h_m é um classificador fraco, e M o número de classificadores, deve-se proceder com os seguintes passos:

1. Inicialize a função $h_0(x) = min \sum_{i=1}^{n} L$
2. Cada um dos M classificadores sequenciais são calculados com os seguintes passos:
   a) Obtém-se a função negativa de gradiente $-g = -\frac{\delta L}{\delta F}$ para cada um dos elementos da entrada $[x_1, x_2, ..., x_n]$.
   b) Ajustar o classificador fraco para predizer $-g$.
   c) Calcular a função de perda L de modo a minimizar o erro.

d) Atualizar a função $h_m = h_{m-1} + L$.

3. O passo 2 será repetido k vezes, uma vez para cada uma das k classes constantes no problema em cada uma da m iterações.
4. Ao fim dos M passos a função F, que aproxima a função real F^* será igual a função h_M.

2.3 Regressão

Um segundo tipo de aprendizagem supervisionada é a predição de valores contínuos, o qual também é chamado de análise de regressão. Nesta abordagem, recebemos um número de variáveis preditoras (explicativas) e uma variável de resposta contínua (resultado), e tentamos encontrar uma relação entre essas variáveis que nos permite predizer um resultado (RASCHKA, 2015).

De modo similar à classificação, na regressão também podemos fazer uma associação, no entanto, desta vez com variáveis quantitativas. Alguns exemplos são: peso; idade; renda; valor de um imóvel.

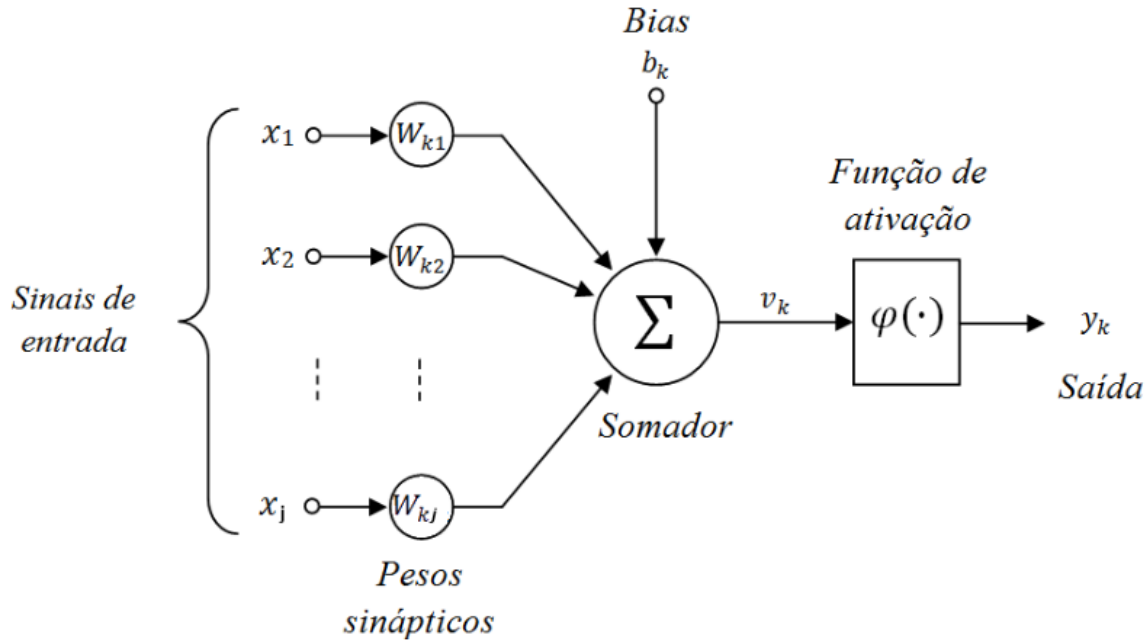
2.3.1 Multilayer Perceptron

Buscando inspiração no cérebro humano, mais especificamente na forma como se dá o processamento das informações através do complexo sistema neuronal, foi criado um modelo matemático computacional capaz de reproduzir comportamento semelhante ao do neurônio humano, tornando possível criar modelos preditivos para problemas que antes não existia tratamento satisfatório.

A Figura 6 apresenta uma ilustração matemática de como se dá o funcionamento de um neurônio artificial. Na primeira parte deste neurônio são apresentados os sinais de entrada $x_i[x_1, x_2, ..., x_j] \in X$ que representam os dados conhecidos de uma base de dados, estes por sua vez são multiplicados pelo vetor de pesos $w_k[w_1, w_2, ..., w_j] \in W$, escolhidos de maneira aleatória, então o resultado desta multiplicação é somado juntamente com o bias b_k (um escalar que tem como função deslocar a fronteira de decisão em relação à origem) gerando como resultado o valor v_k, que é processado em uma função de ativação $\varphi(.)$ para restringir a amplitude de saída y_k do neurônio k, normalmente um valor no intervalo $[0, 1]$ ou $[-1, 1]$, no entanto, especialmente para problemas de regressão, outros valores podem ser assumidos.

De acordo com (MICHIE *et al.*, 1994) Rosenblatt estudou as capacidades de grupos

Figura 6 – Neurônio Artificial



Fonte: (HAYKIN, 2001).

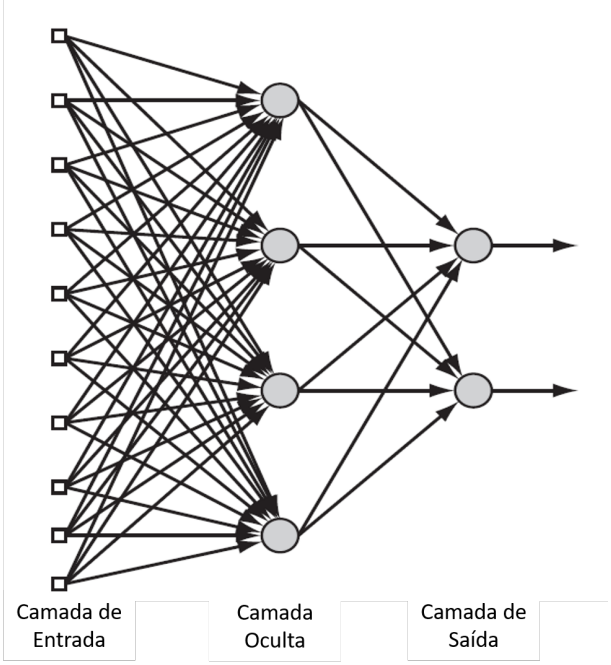
de neurônios em uma única camada, e portanto todos agindo nos mesmos vetores de entrada; esta estrutura foi chamada de perceptron e Rosenblatt propôs a regra de aprendizagem do perceptron para aprender pesos adequados para problemas de classificação. Diante disso e das definições acerca de neurônios artificiais, podemos definir o perceptron como sendo $y_k = \varphi(b_k + \sum_i w_{ki} X_i)$, no qual a saída é um valor binário para resolver problemas de classificação, porém podemos fazer uma alteração na função de ativação para tornar possível a resolução de problemas de regressão.

No entanto, vale ressaltar que a proposta do perceptron não resolve problemas não lineares, onde vários domínios modernos se enquadram, logo uma grande quantidade de problemas não teriam um resultado satisfatório, então (MINSKY; PAPERT, 1969) propôs a utilização de mais de uma camada para resolver tais problemas, o Perceptron de Multicamadas.

A Figura 7 mostra o funcionamento de uma rede neural genérica do tipo MLP com uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Com essa arquitetura se torna possível a resolução de problemas não lineares, pois os dados são processados e as saídas são repassadas como entrada para a camada seguinte até que chegue a camada de saída e produza um resultado, desse modo sempre fazendo a transformação do espaço disposto na camada anterior, possibilitando a separação das classes e a predição mais precisa de valores.

Os pesos W contidos nas camadas são ajustados de forma supervisionada através de uma técnica de correção conhecida como retropropagação do erro, onde a diferença entre os valores obtidos na camada de saída e os valores reais servem para ajustar os pesos de modo

Figura 7 – Multilayer Perceptron



Fonte: (HAYKIN, 2001).

recursivo para as camadas anteriores com o objetivo de que a resposta da rede se aproxime da resposta real. Portanto o modo de funcionamento do MLP se dá em duas etapas de maneira iterativa, uma para frente para cálculo dos valores e outra para trás ajustando os pesos.

2.4 Resumo do Capítulo

Neste capítulo foram apresentados os fundamentos de aprendizagem de máquina diferenciando aprendizado supervisionado, não supervisionado e por reforço no capítulo 2.1. Em seguida foi apresentada uma breve introdução da tarefa de classificação de algoritmos no capítulo 2.2, bem como de maneira simplificada o funcionamento dos algoritmos Support Vector Machine, Gradiente Boosting e Floretas Aleatórias. Por fim, no capítulo 2.3, abordou-se uma introdução sobre algoritmos de regressão, finalizado com a apresentação do modelo de regressão conhecido como Multilayer Perceptron.

3 DADOS FALTANTES

O propósito dos métodos de aprendizado de máquina é essencialmente apoiar a tomada de decisões, no entanto esta tarefa é, por muitas vezes, inviabilizada pela existência de dados faltantes. Os métodos modernos para o tratamento de dados faltantes envolvem a estimação dos valores faltantes. Este processo é usualmente denominado de imputação. De acordo com (LITTLE; RUBIN, 2002) a escolha do método ideal para imputação passa pelo conhecimento dos mecanismos que acarretam em dados faltantes.

3.1 Mecanismos Geradores de Dados Faltantes

De acordo com (LITTLE; RUBIN, 2002), os mecanismos de dados faltantes podem ser caracterizados como Perdas Completamente Aleatórias (*Missing Completely at Random* (MCAR)), Perdas Aleatórias (*Missing at Random* (MAR)) e Perdas Não-Aleatórias (*Not Missing at Random* (NMAR)).

Imagine que temos uma matriz Z que representa os dados, onde cada linha representa um exemplo e as colunas representam os atributos. De mesmo modo temos uma matriz M com a mesma dimensão de Z, tal que, cada célula $M_{ij} = 1$, se Z_{ij} for dado faltante e $M_{ij} = 0$ caso contrário.

Se notarmos que a probabilidade de M é independente dos dados observados e não observados em Z, ou seja, $P(M \mid Z) = P(M)$ então temos MCAR, ou seja, os dados faltantes constituem uma subamostra aleatória. Logo, se pegarmos uma variável qualquer em Z com dado ausente, seu valor não estará relacionado ao de qualquer outra variável no conjunto Z.

Em uma abordagem menos radical, seja Z_{obs} os componentes observados de Z, Z_{mis} os dados faltantes, então a probabilidade de se ter valores faltando depende apenas dos valores observados, ou seja, $P(M \mid Z) = P(M \mid Z_{obs})$. Neste caso denominamos o mecanismo de MAR. Isso ocorre, por exemplo, quando se tem dois atributos A e B, onde a probabilidade de que exista um valor ausente em B depenada dos valores observados em A, mas não no próprio B.

Por fim, se a probabilidade de ocorrência de dados faltantes é dependente dos dados observados e não observados, ou seja, $P(M \mid Z) = P(M \mid Z_{obs}, Z_{mis})$, então denominamos NMAR (LITTLE; RUBIN, 2002). Isso é possível de ocorrer quando, por exemplo, um sensor não detecta uma temperatura abaixo de um certo limite ou uma pessoa não informa sua renda em um formulário se seus ganhos anuais forem maiores que um determinado valor (LAKSHMINARAYAN *et*

al., 1999).

3.2 Algoritmos de Imputação

Conforme dito anteriormente, ao longo dos anos, diversos métodos de imputação foram propostos. Uma das alternativas mais simples é utilizar a média aritmética de um atributo para preencher as lacunas, pois essa abordagem se mostra muito fácil de implementar e rápida na execução, além de requerer um custo computacional baixo se comparado com outros métodos.

Outra alternativa para imputação é utilizar métodos baseados em Análise de Componentes Principais (*Principal Components Analisys* (PCA)). Nesse contexto consideramos três métodos para este trabalho: Algoritmo Iterativo (Iterative Algoritm - IA); método Regressão de Dados Conhecidos (known Data Regression - KDR); Algoritmo modificado Iterativo não Linear de Mínimos Quadrados Parciais (Modified Nonlinear Iterative Partial Least Squares Algorithm - modified NIPALS). A seguir é apresentado um algoritmo base para IA, KDR e NIPALS modificado.

Um procedimento comum para construir um modelo PCA de uma base de dados X é através do algoritmo IA, figura 8, que consiste em preencher os dados faltantes com um valor inicial (normalmente zero, embora outros valores como a média dos valores conhecidos do atributo possam ser imputados), produzindo uma base de dados reconstruída da qual o modelo de PCA é ajustado. Substituindo os valores que originalmente estão ausentes pelos dados preditos pelo PCA, uma nova base de dados é obtida e um novo modelo de PCA pode ser ajustado(FOLCH-FORTUNY *et al.*, 2015). O algoritmo é encerrado quando a diferença entre a base de dados ajustada pelo PCA e a base de dados anterior for menor que um limiar pre-estabelecido.

O método KDR apresenta uma pequena mudança em relação ao método anterior, pois um escore é calculado para ajustar os valores de cada imputação feita pelo PCA baseado no método de mínimos quadrados. A ideia deste método é estimar o escore de um novo indivíduo da base de dados de treinamento X, assumindo que as mesmas variáveis são ausentes em cada linha da matiz de dados X (ARTEAGA; FERRER, 2002). Desse modo se considerarmos o algoritmo da figura 8 apenas fazemos a inclusão do seguinte trecho após a linha 6 para estimar cada dado perdido da seguinte forma para cada instância i com dados faltantes:

$$R_{(mis)i} \leftarrow X^T_{(miss)} X_{(obs)} \left(X^T_{(obs)} X_{(obs)}\right)^{-1} X_{obs} R_{(obs)i}$$

Figura 8 – Algoritmo de imputação IA. na linha 1 uma matriz Z é gerada a partir da multiplicação ponto a ponto do complemento de M por X. Na linha 5 um modelo PCA é gerado a partir de X. Na linha 6 uma base aproximada de X é gerada baseada no PCA e na linha 7 uma nova matriz é gerada. Adaptado de (FOLCH-FORTUNY *et al.*, 2015)

Entrada: Uma matriz com dados faltantes X; $\overline{M}$ - A matriz complementar de M;
Saída: Matriz X com os dados faltantes imputados.

1: $Z \leftarrow \overline{M} o X$
2: $t \leftarrow 0$
3: $X_t \leftarrow Z$
4: **repeat**
5: $\quad PCA \leftarrow T_t P_t^T$
6: $\quad R_t \leftarrow 1_n m_t^T + PCA$
7: $\quad X_{t+1} \leftarrow Z + \overline{M} o R_t$
8: $\quad t = t + 1$
9: **until** $X_t - X_{t-1} > limiar$
10: **return** X_t

Outro método de imputação baseado em PCA é o NIPALS modificado. Esse método foi desenvolvido baseado em outro de mesmo nome que é utilizado para lidar com dados faltantes na construção do modelo PCA. Considerando o algoritmo da figura 8 a única mudança que se faz é na linha 5 para obtenção do PCA utilizando o algoritmo NIPALS. Este modelo consiste em uma regressão linear das colunas de uma base de dados X no vetor de escore para obter o vetor de carregamentos, seguido de uma regressão linear nas linhas de X no vetor de carregamentos para se obter uma nova estimativa de escore. A convergência é alcançada quando a média quadrática da mudança do score fica abaixo de um limiar preestabelecido. Quando dados em alguma linha ou coluna estão faltando, a regressão iterativa é feita usando os dados conhecidos apenas e os dados faltantes são ignorados (NELSON, 2002). De posse do PCA os dados são imputados em X.

Existem ainda métodos baseados em estratégias de vizinhos mais próximos. O algoritmo K Vizinhos Mais Próximos Para Caso Incompleto (Incomplete-Case K Nearest Neighbor Imputation - ICKNNI) procura os k vizinhos mais próximos de um determinado dado com atributo faltando e então imputa a média dos valores dos atributos dos k vizinhos mais próximos. O detalhe a ser observado nesse algoritmo é que os k-vizinhos mais próximos também podem ter dados faltantes, desde que estes sejam em atributos distintos da instância que está se imputando (HULSE; KHOSHGOFTAAR, 2011). Portanto o valor a ser imputado em um dado não observado é:

$$Z_{ij} = \sum_{Z_i \in S_{ij}} \frac{Z_{jl}}{k},$$

onde S_{ij} é o conjunto das k instâncias mais próximas de Z_i de acordo com o critério anteriormente citado. l é o índice pertencente ao conjunto de índices observáveis de Z_i.

3.3 Resumo do Capítulo

Neste capítulo foi feita uma abordagem geral sobre dados faltantes com ênfase em um primeiro momento nos mecanismos geradores MCAR, MAR e NMAR. No subtópico seguinte foi tratado dos algoritmos de imputação utilizados para este estudo: imputação da Média; KDR; NIPALS; IA; ICKNNI.

4 META-APRENDIZADO

4.1 Conceitos Gerais

No estado da arte atual temos dezenas ou até centenas de algoritmos para as mais variadas atividades de aprendizado de máquina tais como classificação, regressão, agrupamento. No entanto, as técnicas aplicadas com ótimos resultados em uma determinada base de dados têm apenas desempenho regular em outro domínio, o que torna difícil escolher o melhor algoritmo toda vez que se depara com uma base de dados nova.

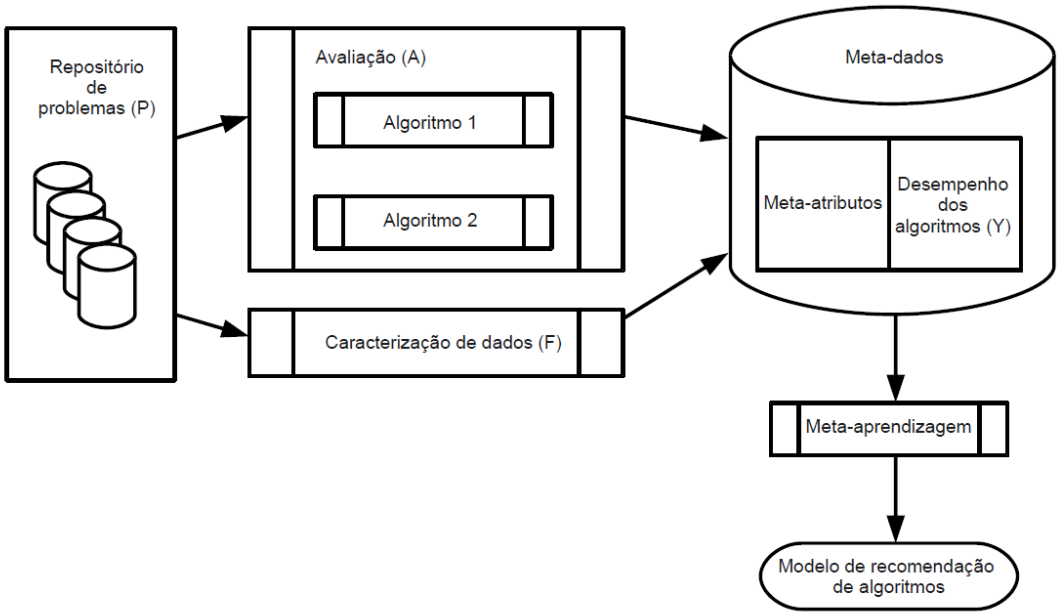
Uma solução para escolha de melhor algoritmo seria testar todos eles na base de dados que se almeja extrair informações e fazer predições, contabilizar o desempenho com metodologia apropriada e finalmente escolher aquele que atende as necessidades. O problema dessa solução é que isso pode levar muito tempo, especialmente para bases de dados grandes e geograficamente distribuídas, inviabilizando o projeto. A solução mais adequada, portanto, é utilizar um sistema de recomendação de algoritmos eficiente.

(BRAZDIL *et al.*, 2009) nos apresenta a distinção entre aprendizado base (svm, random forest, knn) e meta-aprendizado, onde o primeiro foca em acumular experiência em uma tarefa de aprendizado específica, já o aprendizado no nível meta se concentra em acumular experiência na performance de múltiplas aplicações do sistema de aprendizagem.

A proposta do meta-aprendizado dentro do contexto de aprendizagem de máquina é desenvolver modelos que sirvam para para recomendar algoritmos (abordagem adotada neste trabalho), escolher os hiperparâmetros mais adequados, utilizar a melhor estratégia, entre outros. Para isso pressupõe-se que existem bases de dados avaliadas e desenvolvidas por especialistas com o auxílio de aprendizagem de máquina, portanto esse conhecimento prévio é utilizado para desenvolver o modelo de meta-aprendizado.

A Figura 9 apresenta de maneira simplificada como se dá o processo de meta-aprendizado. No primeiro passo obtém-se um repositório de problemas que vão servir de base para extração do conhecimento. Em seguida para cada base de dados algoritmos de machine learning são rodados para se obter um ranqueamento do desempenho dos mesmos. Por outro lado, cada uma das bases são caracterizadas de acordo com os meta-atributos (número de linhas, número de colunas, média, entre outros) pré-selecionados. De posse dos meta-atributos e dos ranqueamentos dos algoritmos de cada base, realiza-se a combinação de ambos para formar as instâncias dentro de uma base de meta-dados. Por fim separa-se a base de meta-dados obtida em

Figura 9 – Fluxograma mostrando os passos da seleção de um algoritmo baseado em meta-aprendizado.



Fonte: (SOUZA *et al.*,).

treino e teste para se aplicar um algoritmo com o intuito de obter um modelo consistente para predições de melhor algoritmo para uma base de dados nova.

4.2 Meta-Atributos

Meta-atributo é uma medida calculada sobre uma base de dados para descrever suas propriedades e características. Os meta-atributos constroem o espaço de atributos no qual cada base de dados é representado por um ponto (REIF *et al.*, 2012). Quando temos todos os meta-atributos reunidos podemos utilizá-los para inferir conhecimento como, por exemplo, predizer qual algoritmo de aprendizagem de máquina tem melhor desempenho de classificação em uma determinada base de dados nova, que não foi usada no treinamento.

Para um melhor entendimento da importância dos meta-atributos devemos considerar que este exerce papel central nas tarefas relacionadas a meta-learning, pois são eles que provém informações sobre as bases de dados, essenciais para tomada de decisão. No entanto, um grande desafio encontrado pelos meta-atributos é que normalmente um único valor descreve a bases de dados inteira, o que pode eventualmente conduzir a um mesmo valor para bases de dados completamente diferentes, por isso se torna importante obter um número maior de meta-atributos para descrever uma base de dados e, dessa forma, evitar confusões na tarefa de predição.

4.3 Caracterização de Bases de Dados

Caracterizar bases de dados consiste em identificar e extrair propriedades que, possivelmente, afetem o desempenho dos algoritmos de classificação. O objetivo da caracterização é fornecer informações morfológicas dos dados para a aplicação de técnicas de meta-aprendizado (SOUZA *et al.*,). Atualmente temos três tipos principais de caracterização de bases de dados: caracterização direta, que foi adotada neste trabalho; caracterização via landmarking; caracterização via modelos.

A caracterização direta é uma das formas mais simples de descrever uma base de dados pois considera atributos que podem ser obtidos diretamente da base em questão. Uma das primeiras iniciativas para caracterização direta através de meta-atributos foi feita no projeto Statlog (MICHIE *et al.*, 1994), que os separou em três categorias: simples, onde as informações são coletadas diretamente como, por exemplo, número de linhas, colunas; estatísticos como média, desvio-padrão, curtose; baseados em informação como entropia média dos atributos.

A ideia do landmarking é explorar a informação obtida da performance de um conjunto de modelos simples de aprendizado (sistemas com baixa capacidade por exemplo) que tenham diferenças significativas em seu mecanismo. A acurácia (ou taxa de erro) desses landmarkers é usado para caracterizar a base de dados. O objetivo é identificar áreas no espaço entrada onde cada um desses sistemas simples de aprendizado pode ser considerado como um especialista. Este meta conhecimento pode ser subsequentemente explorado para produzir modelos melhores. Outra ideia relacionada a landmarking é explorar informação obtida em versões simplificadas de dados (MAIMON; ROKACH, 2010).

De modo diferente do anterior, caracterização via modelos ocorre pela exploração da estrutura dos modelos adotados e não pelos desempenho dos algoritmos. Para (SOUZA *et al.*,) A utilização de modelos para a caracterização de bases de dados realiza uma mudança no espaço de busca do algoritmo de meta-aprendizado, passando do espaço de exemplos para o espaço de hipóteses do algoritmo utilizado para a caracterização. Uma escolha comum para este tipo de caracterização é a utilização de árvore de decisão, onde os meta-atributos selecionados são, por exemplo, o número de nós, a profundidade máxima, a forma.

4.4 Resumo do Capítulo

Neste capítulo foram discutidas as bases gerais referentes a meta-aprendizagem, passando pelas definições com a apresentação de um modelo geral para recomendação de algoritmos no capítulo 4.1. No capítulo 4.2 foi feita uma abordagem conceitual sobre os meta-atributos. Por fim, no capítulo 4.3 foram discutidos três estratégias para caracterização de bases de dados: caracterização direta; caracterização por landmanking; caracterização via modelos.

5 METODOLOGIA

A metodologia utilizada neste trabalho pode ser dividida em duas etapas com objetivos distintos. Na primeira etapa foi feito um estudo para verificação do impacto de diferentes métodos de imputação no desempenho de métodos de aprendizagem de máquina em tarefas de classificação e regressão. A segunda etapa consiste em um sistema de recomendação de algoritmos de imputação baseado em conceitos de meta-aprendizado.

5.1 Impacto de Métodos de Imputação

Para esta etapa do trabalho foram usados 42 conjuntos de dados, dos quais 22 são referentes a problemas de regressão e 20 são referentes problemas de classificação, com três percentuais distintos de dados faltantes, 5%, 35% e 60%, gerados artificialmente, da seguinte forma: para cada instância de cada base de dados, foram apagados até no máximo a metade dos atributos, escolhidos de maneira aleatória, de modo que exista pelo menos um atributo com dado faltante em 5%, 35% ou 60% das instâncias. Essa abordagem se deu para que fosse possível abranger uma quantidade maior de situações distintas envolvendo problemas de aprendizagem de máquina.

O mecanismo utilizando para gerar os dados faltantes foi o MCAR, haja vista essa abordagem ser mais simples de implementar e ter uma ampla utilização em periódicos. Sendo assim, podemos apagar dados aleatoriamente sem entrar nos detalhes específicos de cada base de dados.

Depois de gerados os dados faltantes em cada uma das bases de dados listados na Tabela 1 com seus respectivos percentuais de dados faltantes, foram aplicados os métodos de imputação da média, KDR, NIPALS, IA e ICKNNI para gerar uma base de dados completa. Esses métodos foram escolhidos por conta da disponibilidade no MDI Toolbox e pela facilidade de implementação no caso do ICKNNI e da média, sem que isso implicasse na perda de qualidade dos métodos.

Em seguida, de modo a abranger diferentes domínios de aprendizagem de máquina, foi aplicado Perceptron de Multicamadas (Multilayer Perceptron - MLP) para os problemas de regressão e foi utilizado Máquina de Vetor de Suporte (Support Vector Machine - SVM) (RUSSELL; NORVIG, 2012) para os problemas de classificação. Esse processo se repetiu por 10 vezes para cada base de dados para obtenção da média dos resultados de acurácia (erro quadrático

médio para regressão e percentual de erro para classificação) dos métodos, evitando desse modo que fossem produzidos resultados tendenciosos ou inapropriados.

A significância estatística dos resultados de cada método foi testada de acordo com o teste de Friedman (FRIEDMAN, 1937) que utiliza o ranking médio de cada método. Desse modo é possível avaliar o quanto cada método de imputação impacta no desempenhos dos algoritmos de aprendizagem de máquina. Nos casos onde a a Hipótese nula foi rejeitada foi calculado a diferença crítica (DEMSAR, 2006) para descobrir quais dos métodos apresentam diferença significativa. A Figura 10 mostra o fluxograma desta primeira parte da metodologia.

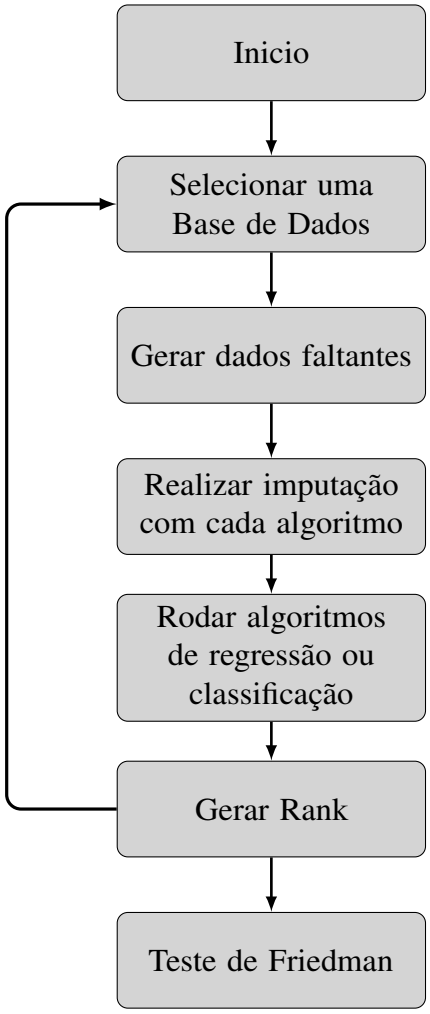


Figura 10 – Fluxograma mostrando os passos para avaliação do impacto de métodos de imputação sobre métodos de aprendizagem de máquina.

5.2 Recomendação de métodos de imputação

Para recomendação de métodos de imputação foi gerado para cada base de dados um percentual aleatório de dados faltantes, variando de 10% a 70%, também utilizando o mecanismo

Tabela 1 – Descrição das Bases de Dados

Base de Dados	**Tamanho**	**Nº de Atributos**	**Tipo**
Airfoil	1503	5	Regressão
Automobile	204	25	Regressão
CBM	11934	17	Regressão
CCPP	9567	4	Regressão
Compression	1030	8	Regressão
CPU	209	9	Regressão
DDFO	60	12	Regressão
Facebook	500	18	Regressão
Forest Fire	517	4	Regressão
Housing	506	13	Regressão
Hydrodinamics	308	6	Regressão
MPG	592	7	Regressão
Parkinson	5875	21	Regressão
Sinc	800	100	Regressão
Slump1	103	8	Regressão
Slump2	103	8	Regressão
Slump3	103	8	Regressão
Transcoding	68783	21	Regressão
Wine Red	1599	11	Regressão
Wine White	4898	11	Regressão
SPP	63	6	Regressão
Servo	167	4	Regressão
Brain	42	5597	Classificação
Breast	569	30	Classificação
Diabetes	768	8	Classificação
Digit	1593	256	Classificação
Ecoli	336	7	Classificação
Glass	214	10	Classificação
Hayes	160	3	Classificação
Heart	270	13	Classificação
Iris	150	4	Classificação
Leaf	340	14	Classificação
Liver	345	6	Classificação
Monk1	556	6	Classificação
Monk2	601	6	Classificação
Monk3	554	6	Classificação
Sonar	208	60	Classificação
Spambase	4601	57	Classificação
Waveform	5000	21	Classificação
Wine	178	13	Classificação
YaleFaces	164	105	Classificação
Flare	1389	11	Classificação

MCAR, apagados de acordo com o tópico anterior da metodologia, pelos mesmos motivos expostos anteriormente.

Na sequência foram extraídos os seguintes meta-atributos para cada uma das bases de dados: número de linhas; número de atributos; percentual de dados faltantes; valor médio das médias dos atributos; valor médio dos desvios padrões dos atributos; correlação absoluta média dos atributos; desvio padrão da correlação absoluta dos atributos; assimetria média dos atributos; desvio padrão da assimetria dos atributos; curtose média dos atributos; desvio padrão da curtose dos atributos; indicação se a base de dados trata de regressão ou classificação; e por fim a acurácia da aplicação dos métodos MLP e SVM sobre as bases de dados, para os problemas de regressão e classificação respectivamente. Esses meta-atributos foram escolhidos devido suas citações na literatura e relevâncias na produção dos resultados para este estudo, no entanto nenhum estudo adicional foi feito para no sentindo de determinar o grau de importância de cada um.

De posse de todos esses meta-atributos, os dados foram separados em treinamento e teste, então aplicou-se o método de classificação Florestas Aleatórias, Máquina de Vetor de Suporte e Gradiente Boosting para predizer o melhor algoritmo de imputação de dados, dentre os que foram anteriormente citados, pois dessa forma obtém-se a classe de cada instância de meta-dados. Este processo foi repetido 10 vezes e então a acurácia média, bem como a matriz de confusão foram extraídas para análise dos resultados. Todo o processo para obtenção dos resultados está exposto na Figura 11.

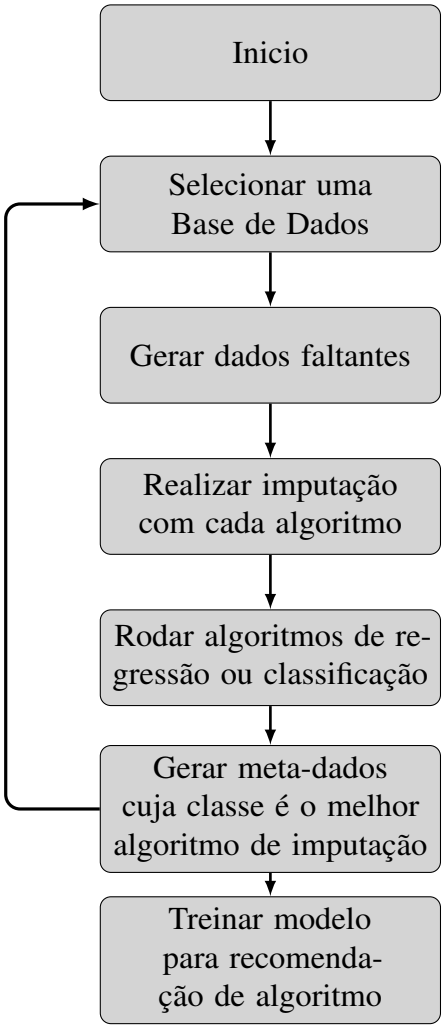


Figura 11 – Fluxograma mostrando os passos para construção de um sistema de recomendação de algoritmos de imputação.

6 RESULTADOS

6.1 Análise do impacto de métodos de imputação

Nessa primeira etapa para analisar os resultados foi utilizado o teste Friedman que avalia a consistência dos resultados obtidos pelos métodos quando aplicados sobre várias bases de dados de acordo com o seu posicionamento de desempenho, ou seja, para cada base de dados o algoritmo que obteve o melhor desempenho é dado a posição 1, o segundo melhor recebe a posição 2 e assim por diante até que todos os algoritmos tenham uma posição. No caso de dois ou mais algoritmos terem o mesmo desempenho, eles recebem a média de posicionamento que teriam se o desempenho fosse diferente, sendo assim se os três melhores algoritmos tiverem o mesmo desempenho, por exemplo, então o posicionamento médio será 2 $((1+2+3)/3 = 2)$.

Para fins deste teste é calculada a média do posicionamento dos algoritmos nas bases de dados, tal qual é mostrado nas Tabelas 2, 3, 4, 5, 6 e 7. Na atual configuração a hipótese nula H_0 é que não existe diferença estatística entre os modelos de imputação de dados (a diferença entre o posicionamento médio não é significativa) com um nível de significância de 5%.

Seguindo a metodologia proposta por (DEMSAR, 2006), não podemos rejeitar a hipótese nula H_0 para Tabela 2, referente aos algoritmos de imputação para problemas de regressão com 5% de dados faltantes, pois o valor-p obtido foi de 0.3665, embora seja possível notar que há uma diferença de desempeno entre os algoritmos, sendo a maior entre ICKNNI (2.86) com melhor desempenho e a Média (3.18) com o pior. Podemos observar ainda que os algoritmos NIPALS e IA tiveram o mesmo desempenho considerando a posição média ficando empatados na terceira posição. Para o KDR restou o segundo melhor desempenho. O resultado similar obtidos por esses algoritmos pode ser justificado pelo baixo percentual de dados faltantes o que torna as bases de dados muito similares sendo imputadas com algoritmos diferentes.

Por outro lado, quando se analisa o desempenho de algoritmos de imputação na tarefa de regressão expostos na Tabela 3, para uma taxa de 35% de dados faltantes, é possível verificar que a hipótese nula H_0 deve ser rejeitada, pois o valor-p é igual a 8.3972e-5. Isso significa que pelo menos dois dos algoritmos são significativamente distintos, então deve-se proceder com a investigação para descobrir quais destes algoritmos apresentam tal distinção. Para isso utiliza-se o cálculo da diferença crítica (*Critical Difference* (CD)) que é calculada de acordo com a proposta de (DEMSAR, 2006).

Tabela 2 – Posição Média - Regressão - 5% Dados Faltantes

	MÉDIA	KDR	NIPALS	IA	ICKNNI
Airfoil	4	1	5	3	2
Automobile	1	3	2	5	4
CBM	3	1	5	4	2
CCPP	3	1	4	5	2
Compression	5	1	3	4	2
CPU	1	2	5	3	4
DDFO	3	1	4	2	5
Facebook	5	3	2	4	1
Forest Fire	5	4	1	2	3
Housing	4	3	1	5	2
Hydrodinamics	5	1	4	3	2
MPG	2	3	5	4	1
Parkinson	1	4	3	2	5
Sinc	4	5	3	2	1
Slump1	3	5	1	2	4
Slump2	4	5	2	1	3
Slump3	3	2	5	4	1
Transcoding	2	4	1	3	5
Wine Red	3	5	2	1	4
Wine White	3	5	1	2	4
SPP	5	4	2	1	3
Servo	1	2	5	4	3
Posição Média	3.18	2.95	3.00	3.00	2.86

Ao se calcular a CD obtém-se valor igual a 1.3005 e de acordo com a metodologia adotada, significa que dois métodos são significativamente diferentes se a posição média entre eles diferir em pelo menos CD. Posto isso, é possível notar que o KDR, melhor modelo, com posição média de 1.89, é significativamente diferente da Média, superando-a em 1.34, é significativa diferente do método NIPALS, superando-o em 1.61, e é significativamente diferente do método IA, superando-o em 1.97, no entanto não apresenta diferença significativa para o ICKNNI, este por sua vez é significativamente melhor que o método IA, superando-o em 1.34. Entre quaisquer outros pares não existe diferença significativa.

Posto isso é possível notar que para 35% de dados faltantes, o método de imputação escolhido impacta significativamente no modelo de regressão adotado se determinados pares de algoritmos de imputação forem comparados.

De maneira similar ao que foi observado na análise de 5% de dados faltantes, não podemos rejeitar a hipótese nula H_0 para Tabela 4, tratando de imputação de dados em problema

Tabela 3 – Posição Média - Regressão - 35% Dados Faltantes

	MÉDIA	KDR	NIPALS	IA	ICKNNI
Airfoil	5	1	3	4	2
Automobile	5	1	4	2	3
CBM	2	1	5	4	3
CCPP	3	1	4	5	2
Compression	5	1	3	4	2
CPU	5	2	3	4	1
DDFO	2	3	4	1	5
Facebook	2	3	1	5	4
ForestFire	1	2.5	4	5	2.5
Housing	2	3	5	4	1
Hydrodinamics	2	3	1	4	5
MPG	5	1	3	4	2
Parkinson	5	1	4	3	2
Sinc	2	5	4	3	1
Slump1	3	1	4	5	2
Slump2	1	3	4	5	2
Slump3	2	1	4	3	5
Transcoding	4	1	5	3	2
WineRed	5	2	1	4	3
WineWhite	5	1	3	4	2
SPP	1	3	5	4	2
Servo	4	1	3	5	2
Posição Média	3.23	1.89	3.50	3.86	2.52

de regressão, no entanto com 60% de dados faltantes e valor-p = 0.0912. É possível notar ainda que a diferença entre a melhor e a pior posição média aumenta com 60% de dados faltantes, sendo ICKNNI e KDR igual a 2.45 e IA 3.45, portanto apesar de haver um aumento na diferença entre os algoritmos quando comparamos com 5% de dados faltantes, esta continua não significativa.

Quando mudados para análise do impacto dos algoritmos de imputação sobre os problemas de classificação é possível perceber que para 5% de dados faltantes, Tabela 5, a hipótese nula H_0 não pode ser rejeitada, pois o valor-p é igual 0.7265. No entanto quando se observas a Tabela 5 percebe-se que apesar de o algoritmo ICKNNI aprensentar um melhor desempenho se considerar a posição média, não há destaque de nenhum algoritmos pois há empate em quase todos as bases de dados, indicando que 5% de dados faltantes não é um valor muito relevante a ponto de o algoritmo de imputação impactar na tarefa de classificação.

Nesse mesmo sentido, podemos observar que para 35% de dados faltantes, Tabela 6, a hipótese nula H_0 não pode ser rejeitada, haja vista o valor-p ser de 0.7207. No entanto já é

Tabela 4 – Posição Média - Regressão - 60% Dados Faltantes

	MÉDIA	KDR	NIPALS	IA	ICKNNI
Airfoil	5	1	3	4	2
Automobile	5	3	1	4	2
CBM	2	1	5	4	3
CCPP	3	2	4	5	1
Compression	3	1	4	5	2
CPU	3	5	2	1	4
DDFO	3	5	2	1	4
Facebook	4	1	2	3	5
Forest Fire	1	3	4	5	2
Housing	1	2	5	4	3
Hydrodinamics	3	4	2	1	5
MPG	5	1	3	4	2
Parkinson	5	1	4	3	2
Sinc	4	5	2	3	1
Slump1	1	5	4	3	2
Slump2	4	5	2	3	1
Slump3	5	1	4	3	2
Transcoding	5	1	3	4	2
Wine Red	1	3	5	2	4
Wine White	4	1	3	5	2
SPP	3	1	4	5	2
Servo	3	2	5	4	1
Posição Média	3.32	2.45	3.32	3.45	2.45

possível notar que o algoritmo de imputação já ganha uma importância bem maior, pois ocorre uma quantidade significativamente menor de empate de desempenho quando se analisa cada uma das instâncias. O algoritmo de imputação da média merece o destaque, pois apesar de ter aumentado o percentual de dados faltantes, o mesmo teve o melhor desempenho, diferentemente do esperado para um algoritmo tão simplista, mesmo quando se trata de problemas de classificação.

Por fim, quando se analisa o impacto de 60% de dados faltantes sobre o desempenho do classificador máquina de vetores de suporte, demonstrado na Tabela 7, com valor-p igual a 0.6891, conclui-se que não é possível rejeitar a hipótese nula H_0, no entanto é notório que não há aumento na diferença entre a maior e a menor posição média quando se compara 5% e 60% de dados faltantes, apesar de haver um discreto aumento quando se compara 35% para classificação e assim como em algumas das análises anteriores os métodos ICKMMI e KDR prevalecem como os melhores.

É importante notar que, apesar da hipótese nula poder ser rejeitada em apenas uma

Tabela 5 – Posição Média - Classificação - 5% Dados Faltantes

	MÉDIA	KDR	NIPALS	IA	ICKNNI
Brain	3	3	3	3	3
Breast	1	2	4	4	4
Diabetes	5	2.5	2.5	2.5	2.5
Digit	2.5	5	2.5	2.5	2.5
Ecoli	3	3	3	3	3
Glass	3	3	3	3	3
Hayes	2	3	5	1	4
Heart	2	4.5	4.5	2	2
Iris	2.5	2.5	2.5	5	2.5
Leaf	3	3	3	3	3
Liver	2	2	4.5	4.5	2
Monk1	3	3	3	3	3
Monk2	3	3	3	3	3
Monk3	3	3	3	3	3
Sonar	2.5	5	2.5	2.5	2.5
Spambase	3.5	3.5	3.5	3.5	1
Waveform	4	5	3	2	1
Wine	3.5	3.5	3.5	3.5	1
YaleFaces	2	4	1	3	5
Flare	3	3	3	3	3
Posição Média	2.83	3.32	3.15	3.00	2.70

das seis situações testadas, 35% de dados faltantes para regressão, o impacto do uso de diferentes métodos de imputação é maior em problemas de regressão quando se analisa a posição média. Este fator pode ser explicado pelo fato do problema de regressão ser usualmente mais complexo, pois é necessário estimar o valor contínuo exato da saída, enquanto que nos problemas de classificação procura-se predizer um número discreto pequeno de classes. Outra observação interessante é que a diferença de desempenho entre os métodos é maior quando se faz a imputação de 35% de dados faltantes.

6.2 Recomendação de métodos de imputação

Após a análise feita no tópico 6.1 foi possível perceber que, com exceção do ocorrido com 35% de dados faltantes para regressão, os métodos de imputação não tiveram impacto significativamente diferente nos algoritmos de classificação e regressão. Isso motivou a segunda parte da análise, pois foi possível notar que quando se mudava as características da base de dados também mudava o algoritmo de imputação com melhor desempenho. Logo, este se tornou um

Tabela 6 – Posição Média - Classificação - 35% Dados Faltantes

	MÉDIA	KDR	NIPALS	IA	ICKNNI
BrainDB	3	3	3	3	3
Breast	4	1	4	4	2
Diabetes	4.5	4.5	1	2.5	2.5
Digit	1	5	3	2	4
Ecoli	3.5	3.5	3.5	3.5	1
Glass	4	2	5	2	2
Hayes	5	4	2	1	3
Heart	1	2	5	4	3
Iris	1.5	4	1.5	4	4
Leaf	1	3	5	4	2
Liver	4	1.5	5	3	1.5
Monk1	2.5	2.5	2.5	2.5	5
Monk2	3	3	3	3	3
Monk3	2	2	5	2	4
Sonar	2	1	4	4	4
Spambase	2	2	5	2	4
Waveform	3.5	5	1	2	3.5
Wine	3	3	3	3	3
YaleFaces	1	4	3	5	2
Flare	3	3	3	3	3
posição Média	2.73	2.95	3.38	2.98	2.98

cenário ideal para o desenvolvimento de um modelo para recomendação de algoritmos, cuja análise é feita a partir de agora.

A segunda parte da análise é referente à aplicação de Máquina de Vetor de Suporte e algoritmos de conjuntos, Florestas Aleatórias e Gradiente Boosting, para predizer o melhor algoritmo de imputação baseado nos meta-atributos previamente coletados de acordo com a metodologia para cada base de dados. Para avaliar o resultado foi utilizado a média da medida F de acordo com (FERRI *et al.*, 2009). Para tal, foram realizadas classificações dos meta-dados 10 vezes no grupo de treinamento e então foram obtidos os seguintes resultados:

Da aplicação do Gradiente Boosting para predizer o melhor algoritmo de imputação observou-se ao analisar o relatório de classificação disposto na Tabela 8 Precisão Média foi de 0.96, mostrando que existem na amostragem apenas 4% de falsos negativos, por outro lado a Revocação Média apresentou um valor de 0.92, ou seja, temos apenas 8% de falsos negativos. Já a Medida F nos traz a média harmônica entre a precisão e a revocação cujo valor é de 0.93.

Analisando a matriz de confusão contida na Tabela 9 ainda com o algoritmo de

Tabela 7 – Posição Média - Classificação - 60% Dados Faltantes

	MÉDIA	KDR	NIPALS	IA	ICKNNI
Brain	3	3	3	3	3
Breast	1	3.5	3.5	2	5
Diabetes	5	3.5	1	2	3.5
Digit	2.5	2.5	4	5	1
Ecoli	3.5	3.5	1	3.5	3
Glass	3	1	5	3	3
Hayes	3.5	3.5	5	1	2
Heart	5	1	3	4	2
Iris	3	3	3	3	3
Leaf	3	1.5	4	5	1.5
Liver	4.5	3	1	4.5	2
Monk1	2	2	5	2	4
Monk2	3	3	3	3	3
Monk3	2	2	5	4	2
Sonar	5	2.5	1	2.5	4
Spambase	2	3	5	1	4
Waveform	4	5	3	2	1
Wine	5	3.5	3.5	1.5	1.5
YaleFaces	3	1	5	4	2
Flare	4	4	1.5	1.5	4
Posição Média	3.35	2.75	3.28	2.88	2.75

Tabela 8 – Relatório de Classificação para o algoritmo Gradiente Boosting quando aplicado na tarefa de recomendação de melhor modelo de imputação

Modelo	**Precisão Média**	**Revocação Média**	**Medida F**	**Suporte**
MÉDIA	1.00	1.00	1.00	7
KDR	1.00	0.85	0.92	27
NIPALS	0.50	1.00	0.67	1
IA	0.57	1.00	0.73	4
ICKNN	1.00	1.00	1.00	11
média/total	0.96	0.92	0.93	50

Tabela 9 – Matriz de Confusão - Gradiente Boosting

	MÉDIA	KDR	NIPALS	IA	ICKNNI
MÉDIA	7	0	0	0	0
KDR	0	23	1	3	0
NIPALS	0	0	1	0	0
IA	0	0	0	4	0
ICKNN	0	0	0	0	11

Tabela 10 – Relatório de Classificação para o algoritmo Florestas Aleatórias quando aplicado na tarefa de recomendação de melhor modelo de imputação

Modelo	**Precisão Média**	**Revocação Média**	**Medida F**	**Suporte**
MÉDIA	1.00	1.00	1.00	7
KDR	1.00	0.72	0.84	32
NIPALS	0.00	0.00	0.00	2
IA	0.29	1.00	0.44	2
ICKNN	0.64	1.00	0.78	7
média/total	0.88	0.78	0.80	50

Gradiente Boosting é possível notar que os algoritmos de Média e ICKNNI foram perfeitamente preditos sem nenhum equívoco, já quando o KDR foi predito como melhor algoritmo não houve nenhum falso negativo, no entanto ocorreram 4 falsos positivos. Para os algoritmos NIPALS e IA ocorreram predições de 1 e 4 falsos negativos, respectivamente.

Fazendo a análise do algoritmo de Florestas Aleatórias para predizer o melhor modelo de imputação observa-se ao analisar o relatório de classificação disposto na Tabela 10 que Precisão Média apresenta valor de 0.88, o que nos diz que existem na amostragem 12% de falsos negativos, por outro lado a Revocação Média apresentou um valor de 0.78, o que nos leva a concluir que temos 22% de falsos negativos. Já a Medida F nos apresenta um valor de 0.80, logo, se fizermos uma comparação com Gradiente Boosting veremos que há um aumento razoável no erro tanto no percentual de falsos positivos e falsos negativos quanto na média harmônica por consequência, no entanto ainda se mantém um resultado satisfatório. Uma observação importante se faz para NIPALS que apresentou tanto Precisão Média como Revocação Média iguais zero por não ter ocorrido nenhuma predição correta para esta classe.

Analisando a matriz de confusão contida na Tabela 11 para o modelo de Florestas Aleatórias é possível notar que a Média não teve nenhuma predição equivocada, no entanto quando o KDR foi predito como melhor algoritmo não houve nenhum falso negativo, mas ocorreram 9 falsos positivos. Para NIPALS ocorreram 2 falsos positivos e 2 falsos negativos.

Tabela 11 – Matriz de Confusão - Florestas Aleatórias

	MÉDIA	KDR	NIPALS	IA	ICKNNI
MÉDIA	7	0	0	0	0
KDR	0	23	2	3	4
NIPALS	0	0	0	2	0
IA	0	0	0	2	0
ICKNN	0	0	0	0	7

Tabela 12 – Relatório de Classificação para o algoritmo Máquina de Vetor de Suporte quando aplicado na tarefa de recomendação de melhor modelo de imputação

Modelo	**Precisão Média**	**Revocação Média**	**Medida F**	**Suporte**
MÉDIA	0.71	0.83	0.77	6
KDR	0.96	0.69	0.80	32
NIPALS	0.00	0.00	0.00	0
IA	0.00	0.00	0.00	0
ICKNN	1.00	0.92	0.96	12
média/total	0.94	0.76	0.83	50

Já IA teve 2 acertos e 5 falsos negativos. Por fim ICKNNI teve 7 predições corretas e 4 falsos negativos.

Por fim vamos analisar o desempenho do algoritmo Máquina de Vetor de Suporte para predizer o melhor modelo de imputação. Adentrando no relatório de classificação disposto na Tabela 12 que Precisão Média apresenta valor de 0.94, que representa um percentual baixo de falsos positivos na proporção de 6%. De maneira oposta, ao se analisar a Revocação Média com valor de 0.76, percebe-se uma taxa de 24% de falsos negativos, a mais elevada entre os algoritmos analisados. Já a Medida F nos apresenta um valor de 0.83, cujo valor mais elevado é influenciado pelo fato de tanto NIPALS quanto IA não terem sido preditos corretamente nenhuma vez, já que tanto no cálculo da Precisão Média quanto da Revocação Média valores zerados não exercem influencia sobre o total.

Analisando a matriz de confusão contida na Tabela 13 para o modelo de Máquina de Vetor Suporte é possível notar que a Média teve 5 predições corretas, 1 falso positivo e 2 falsos negativos. No caso do KDR houve 22 predições corretas como melhor algoritmo, houve 1 falso negativo e ocorreram 10 falsos positivos. Para NIPALS ocorreram 2 falsos negativos. Já IA teve 7 falsos negativos. Por fim ICKNNI teve 11 predições corretas 1 falsos positivo.

Fazendo uma análise comparativa com os dois algoritmos anteriores nota-se que a Máquina de Vetor de Suporte teve um desemprenho inferior aos modelos anteriores, ainda que

Tabela 13 – Matriz de Confusão - Máquina de Vetor de Suporte

	MÉDIA	KDR	NIPALS	IA	ICKNNI
MÉDIA	5	1	0	0	0
KDR	2	22	2	6	0
NIPALS	0	0	0	0	0
IA	0	0	0	0	0
ICKNN	0	0	0	1	11

tenha uma Medida F mais alta que Florestas Aleatórias, pois apresentou um número maior de predições falsas, ainda que tenha usado kernel RBF.

Se for considerado apenas o percentual de acerto de predição do melhor algoritmo de imputação obtém-se 92% para Gradiente Boosting, 80% para Florestas aleatórias e 76% para Máquina de Vetor de Suporte com kernel RBF. Por outro lado, se for considerado acerto quando a predição indicar como melhor algoritmo o primeiro ou segundo melhor, esse mesmo percentual sobe para 94% para Gradiente Boosting, 84% para Florestas Aleatórias e 82% para SVM, ou seja, se formos considerar apenas o melhor desempenho teríamos para cada 100 indicações de melhor algoritmo de imputação 92 vezes o melhor de fato e 2 vezes o segundo melhor.

7 CONCLUSÕES E TRABALHOS FUTUROS

Nesta dissertação foi proposto analisar o impacto da aplicação de diferentes métodos de imputação sobre dados faltantes no desempenho de algoritmos de aprendizado de máquina, a saber: Perceptron de Multicamadas e Máquina de Vetor de Suporte. Foram utilizados os modelos ICKNNI, Média, IA, NIPALS e KDR para preencher os dados ausentes, estes gerados a partir da abordagem MCAR.

De posse dos resultados computados através de tabelas de posição média para algoritmos de classificação e regressão fez-se a análise utilizando o teste de Friedman, que mostrou haver diferença significativa entre os métodos apenas quando considerado 35% de dados faltantes na tarefa de regressão. Diante deste cenário fez-se uso da Diferença Crítica para saber especificamente quais pares de métodos de imputação apresentavam diferenças. Logo, foi possível perceber que tais pares foram: KDR-Média; KDR-NIPALS; KDR-IA e ICKNNI-IA, onde o primeiro modelo do par sempre apresentou melhor posição média.

Por outro lado, foi possível notar que os algoritmos de imputação exercem menor influência sobre o desempenho de algoritmos de aprendizagem de máquina quando se trata de tarefas de classificação, tornando essa diferença de desempenho pouco significativa, o que proporciona uma alternância de métodos com melhor desempenho em cada uma das bases de dados.

No outro ponto da análise referente à recomendação de algoritmos de imputação notou-se que por meio da caracterização direta é possível obter um resultado satisfatório, no entanto deve-se fazer uma boa seleção de meta-atributos e escolher um classificador que apresente boa acurácia de predição para as bases de dados selecionadas.

Posto isso, foi possível notar que o algoritmo de Gradiente Boosting teve melhor desempenho dentre os algoritmos testados e de acordo com a metodologia proposta foi seguido de Florestas Aleatórias e Máquina de Vetor de Suporte, respectivamente, ao ser aplicado sobre os meta-atributos previamente selecionados das bases de dados constantes na Tabela 1.

Por fim, é importante ressaltar que este trabalho não tem o pretexto de servir como modelo para todo e qualquer sistema de recomendação para algoritmos de imputação, nem esgotar a comparação do impacto dos algoritmos imputação nos métodos de aprendizagem de máquina. Portanto este estudo serviu para mostrar que é possível construir um sistema de recomendação com resultado satisfatório, todavia com os recursos adequados pode ser expandido para atender as necessidades reais das organização com bases de dados variando de algumas

dezenas até milhões de instâncias.

7.1 Trabalhos Futuros

É válido considerar que a quantidade de algoritmos de imputação de dados, de classificação e regressão foram muito limitados para este estudo, devido à reduzida capacidade de hardware, no entanto com os resultados obtidos vislumbra-se um estudo mais aprofundado com um número maior de algoritmos de imputação, regressão e classificação, além de alcançar outros domínios tais como, por exemplo, aprendizado por reforço ou agrupamento de dados.

Outra análise que pode ser feita no futuro é a utilização de outros mecanismos para caracterização de bases de dados como landmarking ou caracterização por modelos, ou até mesmo a combinação de ambas.

REFERÊNCIAS

ARTEAGA, F.; FERRER, A. Dealing with missing data in mspc: several methods, different interpretations, some examples. **Journal of Chemometrics: A Journal of the Chemometrics Society**, Wiley Online Library, v. 16, n. 8-10, p. 408–418, 2002.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006.

BRAZDIL, P.; GIRAUD-CARRIER, C.; SOARES, C.; VILALTA, R. **Metalearning: Applications to Data Mining**. [S.l.]: Springer, 2009.

DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, p. 1–30, 2006.

FERRI, C.; HERNáNDEZ-ORALLO, J.; MODROIU, R. An experimental comparison of performance measures for classification. Pattern Recognition Letters 30, p. 27–38, 2009.

FOLCH-FORTUNY, A.; ARTEAGA, F.; FERRER, A. PCA model building with missing data: new proposals and a comparative study 146. Chemometrics and Intelligent Laboratory Systems, p. 77–88, 2015.

FOLCH-FORTUNY, A.; ARTEAGA, F.; FERRER, A. Missing data imputation toolbox for matlab. Chemometrics and Intelligent Laboratory Systems 154, p. 93–100, 2016.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 32, p. 675–701, 1937.

HARRINGTON, P. **Machine Learninig in Action**. [S.l.]: Manning Publications Co., 2012.

HAYKIN, S. **Redes Neurais–Princípios e Prática**. [S.l.]: Porto Alegre: Bookman, 2001.

HULSE, J. V.; KHOSHGOFTAAR, T. M. Incomplete-case nearest neighbor imputation in software measurement data. Information Sciences 259, p. 596–610, 2011.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112.

LAKSHMINARAYAN, K.; HARP, S. A.; SAMAD, T. Imputation of missing data in industrial databases. Applied intelligence 11(3), p. 259–275, 1999.

LITTLE, R. J. A.; RUBIN, D. B. **Statistical Analysis with Missing Data**. [S.l.]: Wiley-Interscience, 2002.

MAIMON, O.; ROKACH, L. **Data Mining and Knowledge Discovery Handbook**. [S.l.]: Springer, 2010.

MICHIE, D.; SPIEGELHALTER, D.; TAYLOR, C.; CAMPBELL, J. **Machine learning, neural and statistical classification**. [S.l.]: Ellis Horwood, 1994.

MINSKY, M.; PAPERT, S. **Perceptrons**. [S.l.]: Cambridge, MA: MIT Press, 1969.

NELSON, P. R. **The treatment of missing measurements in PCA and PLS models**. Tese (Doutorado), 2002.

PATTERSON, J.; GIBSON, A. **Deep Learning: A Practioner's Approch**. [S.l.]: O'Reilly Media, 2017.

RASCHKA, S. **Python machine learning**. [S.l.]: Packt Publishing Ltd, 2015.

REIF, M.; SHAFAIT, F.; DENGEL, A. Meta-features: Providing meta-learners more information. 35th German Conference on Artificial Intelligence, p. 74–77, 2012.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approch**. [S.l.]: Prentice Hall, 2012.

SOUZA, B. F. de; PRUDÊNCIO, R. B.; CARVALHO, A. de. Meta-aprendizado para recomendação de algoritmos.

SOVILJ, D.; EIROLA, E.; MICHE, Y.; BJORK, K.; NIAN, R.; AKUSOK, A.; LENDASSE, A. Extreme learning machine for missing data using multiple imputations. Neurocomputing 174, p. 220–231, 2016.

SUYKENS, J. A.; SIGNORETTO, M.; ARGYRIOU, A. **Regularization, optimization, kernels, and support vector machines**. [S.l.]: CRC Press, 2014.

TREVOR, H.; ROBERT, T.; JH, F. **The elements of statistical learning: data mining, inference, and prediction**. [S.l.]: New York, NY: Springer, 2009.

ZHANG, Y. M. C. **Ensemble Machine Learning: Methods and Applications**. [S.l.]: Springer, 2012.