

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Recomendação de técnicas para imputação de valores
usando aprendizado de máquina

Cassiano Zaghi de Oliveira



São Carlos – SP

Recomendação de técnicas para imputação de valores usando aprendizado de máquina

Cassiano Zaghi de Oliveira

Orientador: Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho

Monografia final de conclusão de curso apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

Área de Concentração: Aprendizado de Máquina, Dados Ausentes, Imputação de Dados.

USP – São Carlos
Junho de 2018

Oliveira, Cassiano Zaghi de
Recomendação de técnicas para imputação de valores
usando aprendizado de máquina / Cassiano Zaghi
de Oliveira. - São Carlos - SP, 2018.
48 p.; 29,7 cm.

Orientador: André Carlos Ponce de Leon Ferreira
de Carvalho.

Monografia (Graduação) - Instituto de Ciências
Matemáticas e de Computação (ICMC/USP), São Carlos -
SP, 2018.

1. Aprendizado de Máquina.
 2. Dados Ausentes.
 3. Imputação de Dados.
- I. Carvalho, André Carlos
Ponce de Leon Ferreira de. II. Instituto de Ciências
Matemáticas e de Computação (ICMC/USP). III. Título.

Cassiano Zaghi de Oliveira

Recomendação de técnicas para imputação de valores usando aprendizado de máquina

Monografia final de conclusão de curso apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

Trabalho aprovado. São Carlos – SP, 21 de junho de 2018:

**André Carlos Ponce de Leon Ferreira
de Carvalho**
Orientador

Rosana Teresinha Vaccare Braga
Convidado 1

Luciano Carli Moreira de Andrade
Convidado 2

*Dedico este trabalho aos meus familiares,
que sempre me apoaram em todos os momentos da minha vida.*

AGRADECIMENTOS

Em primeiro lugar, agradeço aos meus pais, irmãos e familiares, pois sem eles eu não conquistaria nada do que já conquistei.

Agradeço também aos amigos que fiz durante o curso, por toda ajuda e suporte que me deram, além da companhia nas noites de estudo. Em especial ao Elias Rodrigues por toda sua vontade e disponibilidade em ajudar nos mais diversos problemas.

Por fim, agradeço ao Luís Paulo Garcia por todo o suporte e ao professor André Carlos Ponce de Leon Ferreira de Carvalho pela orientação neste projeto.

“A persistência é o caminho do êxito.”
(Charlie Chaplin)

RESUMO

OLIVEIRA, C. Z.. **Recomendação de técnicas para imputação de valores usando aprendizado de máquina.** 2018. 48 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

A quantidade de dados gerados pela humanidade vem crescendo nos últimos anos, e com isso a necessidade de técnicas para a extração de conhecimento em cima destas grandes bases de dados tende a crescer. Como o conhecimento é extraído dos dados, é necessário que eles tenham uma alta qualidade. Um problema recorrente nestas bases de dados é o de valores ausentes, em que perde-se ou não se registra alguns valores, tornando a informação incompleta. Desta forma, técnicas para imputar valores nestas lacunas vêm sendo estudadas. Estas técnicas utilizam informações sobre os dados presentes no conjunto de dados para descobrir o valor a ser imputado. Com a base completa, algoritmos de Aprendizado de Máquina são utilizados para encontrar regras entre as informações desta base. Uma aplicação bastante comum destes algoritmos é a geração de um modelo classificador, em que novos dados são inseridos e classificados de acordo com as regras do modelo. Como os algoritmos de Aprendizado de Máquina possuem vieses diferentes em relação a análise dos dados para a extração das informações, eles possuem desempenhos distintos para cada base de dados, sendo melhores para algumas, e piores para outras. O mesmo pode ser dito sobre os métodos de imputação. Com isso, quem for utilizar estas técnicas para extrair as informações dos dados que possuem precisa ter um conhecimento básico sobre diversos algoritmos para saber qual se adequa ao seu caso, ou contratar algum especialista, o que é custoso. Sendo assim, este projeto consiste em um sistema de recomendação de técnicas de imputação e de algoritmos de Aprendizado de Máquina para uma base de dados, além de também realizar uma análise sobre o desempenho destas técnicas e algoritmos em relação as características das bases de dados.

Palavras-chave: Aprendizado de Máquina, Dados Ausentes, Imputação de Dados.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 – Exemplo de uma Árvore de Decisão. | 6 |
| Figura 2 – Exemplo da separação de classes no SVM. | 7 |
| Figura 3 – Exemplo de um esquema do algoritmo <i>Random Forest</i> . | 8 |
| Figura 4 – Exemplo de um conjunto de dados com valores ausentes. | 9 |
| Figura 5 – Exemplo de uma classificação usando KNN para $k = 3$. | 11 |
| Figura 6 – Estrutura do sistema de desenvolvido. | 14 |
| Figura 7 – Resultados do AM C4.5 para a base <i>Wine</i> . | 19 |
| Figura 8 – Resultados do AM RF para a base <i>Wine</i> . | 19 |
| Figura 9 – Resultados do AM SVM para a base <i>Wine</i> . | 20 |
| Figura 10 – Resultados do AM C4.5 para a base <i>Contraceptive Method Choice</i> . | 21 |
| Figura 11 – Resultados do AM RF para a base <i>Contraceptive Method Choice</i> . | 21 |
| Figura 12 – Resultados do AM SVM para a base <i>Contraceptive Method Choice</i> . | 22 |
| Figura 13 – Resultado gráfico das análises feitas com a base <i>Molecular Biology (Promoter Gene Sequences)</i> no modo de recomendação. | 23 |
| Figura 14 – Resultado gráfico das análises feitas com a base <i>Vehicle</i> no modo de recomendação | 24 |
| Figura 15 – Resultados do AM C4.5 para a base <i>Diabetes</i> . | 36 |
| Figura 16 – Resultados do AM RF para a base <i>Diabetes</i> . | 37 |
| Figura 17 – Resultados do AM SVM para a base <i>Diabetes</i> . | 37 |
| Figura 18 – Resultados do AM C4.5 para a base <i>Ionosphere</i> . | 38 |
| Figura 19 – Resultados do AM RF para a base <i>Ionosphere</i> . | 38 |
| Figura 20 – Resultados do AM SVM para a base <i>Ionosphere</i> . | 39 |
| Figura 21 – Resultados do AM C4.5 para a base <i>Iris</i> . | 39 |
| Figura 22 – Resultados do AM RF para a base <i>Iris</i> . | 40 |
| Figura 23 – Resultados do AM SVM para a base <i>Iris</i> . | 40 |
| Figura 24 – Resultados do AM C4.5 para a base <i>LED</i> . | 41 |
| Figura 25 – Resultados do AM RF para a base <i>LED</i> . | 41 |
| Figura 26 – Resultados do AM SVM para a base <i>LED</i> . | 42 |
| Figura 27 – Resultados do AM C4.5 para a base <i>Monk's Problems 1</i> . | 42 |
| Figura 28 – Resultados do AM RF para a base <i>Monk's Problems 1</i> . | 43 |
| Figura 29 – Resultados do AM SVM para a base <i>Monk's Problems 1</i> . | 43 |
| Figura 30 – Resultados do AM C4.5 para a base <i>Parkinsons</i> . | 44 |
| Figura 31 – Resultados do AM RF para a base <i>Parkinsons</i> . | 44 |

| | |
|--|----|
| Figura 32 – Resultados do AM SVM para a base <i>Parkinsons</i> | 45 |
| Figura 33 – Resultados do AM C4.5 para a base <i>PC Req.</i> | 45 |
| Figura 34 – Resultados do AM RF para a base <i>PC Req.</i> | 46 |
| Figura 35 – Resultados do AM SVM para a base <i>PC Req.</i> | 46 |
| Figura 36 – Resultados do AM C4.5 para a base <i>TAE</i> | 47 |
| Figura 37 – Resultados do AM RF para a base <i>TAE</i> | 47 |
| Figura 38 – Resultados do AM SVM para a base <i>TAE</i> | 48 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Recursos de <i>software</i> utilizados. | 13 |
| Tabela 2 – Técnicas de imputação utilizadas. | 16 |
| Tabela 3 – Algoritmos de AM utilizados. | 17 |
| Tabela 4 – Bases de dados utilizadas na extração dos dados em modo de análise. | 18 |
| Tabela 5 – Erros obtidos para a base de dados <i>Wine</i> . | 18 |
| Tabela 6 – Erros obtidos para a base de dados <i>Contraceptive Method Choice</i> . | 20 |
| Tabela 7 – Bases de dados utilizadas no modo de recomendação. | 22 |
| Tabela 8 – Resultado da recomendação para a base <i>Molecular Biology (Promoter Gene Sequences)</i> . | 23 |
| Tabela 9 – Resultado da recomendação para a base <i>Vehicle</i> . | 24 |
| Tabela 10 – Informações e análises para bases originais. | 36 |

LISTA DE ABREVIATURAS E SIGLAS

- AM Aprendizado de Máquina
ARFF *Attribute-Relation File Format*
MAR *Missing At Random*
MCAR ... *Missing Completely At Random*
MLR *Machine Learning in R*
NMAR ... *Not Missing At Random*
RF *Random Forest*
SO Sistema Operacional
SVM *Support Vector Machine*

SUMÁRIO

| | | |
|---------|--|----|
| 1 | INTRODUÇÃO | 1 |
| 1.1 | Motivação e Contextualização | 1 |
| 1.2 | Objetivos | 2 |
| 1.3 | Organização | 3 |
| 2 | REVISÃO BIBLIOGRÁFICA | 5 |
| 2.1 | Considerações iniciais | 5 |
| 2.2 | Aprendizado de Máquina | 5 |
| 2.2.1 | Árvore de Decisão | 6 |
| 2.2.2 | Support Vector Machine | 6 |
| 2.2.3 | Random Forest | 7 |
| 2.2.4 | Validação de Classificadores | 7 |
| 2.3 | Dados ausentes | 8 |
| 2.4 | Métodos de imputação | 10 |
| 2.4.1 | Imputação pela Média ou Moda | 10 |
| 2.4.2 | KNN Imputation | 10 |
| 2.4.3 | Random Forest | 11 |
| 2.5 | Considerações finais | 12 |
| 3 | DESENVOLVIMENTO | 13 |
| 3.1 | Considerações iniciais | 13 |
| 3.2 | Projeto | 13 |
| 3.3 | Funcionamento do sistema | 13 |
| 3.4 | Descrição das atividades realizadas | 15 |
| 3.4.1 | Escolha dos algoritmos de AM | 15 |
| 3.4.2 | Escolha dos métodos de imputação | 15 |
| 3.4.3 | Gerador de valores ausentes | 15 |
| 3.4.4 | Imputador de dados | 16 |
| 3.4.5 | Analizador de desempenho | 16 |
| 3.5 | Resultados obtidos | 17 |
| 3.5.1 | Resultados para o modo de análise | 18 |
| 3.5.1.1 | Análise da base Wine | 18 |
| 3.5.1.2 | Análise da base Contraceptive Method Choice | 19 |

| | | |
|--------------------|--|-----------|
| 3.5.2 | <i>Resultados para o modo de recomendação</i> | 22 |
| 3.5.2.1 | <i>Recomendações para a base Molecular Biology (Promoter Gene Sequences)</i> | 22 |
| 3.5.2.2 | <i>Recomendações para a base Vehicle</i> | 23 |
| 3.6 | Dificuldades e limitações | 24 |
| 3.7 | Considerações finais | 25 |
| 4 | CONCLUSÃO | 27 |
| 4.1 | Contribuições | 27 |
| 4.2 | Relacionamento entre o Curso e o Projeto | 27 |
| 4.3 | Considerações sobre o Curso de Graduação | 28 |
| 4.4 | Trabalhos Futuros | 28 |
| REFERÊNCIAS | | 31 |
| APÊNDICE A | RESULTADOS PARA OUTRAS BASES DE DADOS | 35 |

Capítulo 1

INTRODUÇÃO

1.1 Motivação e Contextualização

Com o rápido avanço da tecnologia, tanto em termos de hardware como de software, a quantidade de dados gerados vem aumentando exponencialmente nos últimos anos. Estima-se que apenas nos últimos 5 anos a humanidade tenha gerado cerca de 90% dos dados de toda sua história ([Doncel \(2018\)](#)). Com toda essa quantidade de dados, juntamente com a incapacidade de humanos processarem tudo para tirar o máximo proveito deles, a necessidade de criar ferramentas que fossem capazes de realizar tal tarefa se tornou naturalmente imprescindível.

Aprendizado de Máquina (do inglês, *Machine Learning*) ([Bishop \(2006\)](#)) surge como um campo de pesquisa para otimizar decisões e processar grandes quantidades de dados, detectando padrões e regras entre eles. Com essas informações em mãos, a criação de modelos preditivos se torna possível. Estes modelos recebem novos dados e os classificam de acordo com suas características. As aplicações de algoritmos de Aprendizado de Máquina são extremamente diversas, indo desde sistemas de recomendações ([Portugal, Alencar e Cowan \(2015\)](#)) e detecção de fraudes em transações com cartão de crédito ([Lesot e d'Allonnes \(2012\)](#)) até sistemas para a detecção de câncer ([Souza \(2010\)](#)).

Com o aumento do uso destes algoritmos, uma preocupação que surge é a qualidade dos dados que serão utilizados. Esta preocupação é válida, pois o modelo a ser construído será baseado nestes dados. Assim, antes do desenvolvimento do modelo que o algoritmo de Aprendizado de Máquina construirá com os dados, existe a fase de pré-processamento de dados cujo objetivo é preparar os dados para serem utilizados. Nesta fase, os dados passam por um processo de limpeza no qual inconsistências nos dados são suavizadas ou removidas. Problemas comuns nos dados são ruídos (também conhecidos por *outliers*) e dados ausentes ([Batista \(2003\)](#)).

Dados ausentes podem ocorrer por diversos motivos como erros no aparelho de medida durante a coleta de dados, falhas na transferência dos arquivos que contêm os dados, arquivos corrompidos, omissão de respostas em pesquisas, entre outros. A maioria dos algoritmos de Aprendizado de Máquina não fornecem suporte para o tratamento de dados ausentes e, para contornar este problema, as soluções comumente utilizadas são remover todas as instâncias que possuem valores ausentes ou imputar dados nestas lacunas ([Silva \(2010\)](#)).

A primeira solução não é muito bem vista, pois os dados podem conter informações

importantes e necessárias para a construção de um bom modelo, e reduzir a quantidade deles pode implicar numa pior qualidade do modelo gerado. Desta forma, vem crescendo nos últimos anos o interesse na etapa de pré-processamento de dados para que estas lacunas não estraguem e nem impeçam os algoritmos de Aprendizado de Máquina de funcionarem corretamente e darem bons resultados.

A escolha da técnica para imputar dados é muito importante, pois conjuntos de dados possuem características diferentes de outros, o que faz com que uma técnica se torne mais adequada para algumas bases. Porém, para escolher a melhor técnica a ser utilizada, a pessoa ou o time que estiver desenvolvendo o projeto de Aprendizado de Máquina deverá ter um conhecimento tanto sobre os dados que coletaram quanto sobre as diversas técnicas existentes para imputação de valores, o que torna o processo custoso, além de necessitar de pessoas com um maior grau de conhecimento sobre o tema. Esta mesma dificuldade pode ser observada na escolha do algoritmo de Aprendizado de Máquina que será utilizado. Como os algoritmos possuem vieses diferentes, o mesmo algoritmo pode gerar bons resultados para uma base de dados, porém resultados não satisfatórios em outra base ([Mitchell \(1997\)](#)).

Estas dificuldades podem ser uma barreira ao acesso dos benefícios do Aprendizado de Máquina para pessoas leigas, pois elas podem não ter o interesse ou o conhecimento necessário para aprender os detalhes de tudo o que devem fazer para construir um modelo, apenas gostariam de criar ou utilizar um que atenda suas necessidades.

1.2 Objetivos

Este projeto tem como objetivo construir um sistema que faz análise e recomendações de algoritmos de Aprendizado de Máquina e de métodos de imputação de dados ausentes para determinadas bases de dados. A análise pode ser feita tanto para bases de dados completas como para bases de dados que já contenham valores ausentes.

No caso do uso para bases de dados completas, o sistema gera várias bases com diversas porcentagens de valores ausentes, imputa dados nestas bases com diferentes técnicas e gera modelos preditivos usando algoritmos de Aprendizado de Máquina. Com isso, uma comparação entre o desempenho dos diferentes algoritmos pode ser feita, assim como qual método de imputação forneceu o melhor resultado para o algoritmo em questão.

No caso do uso para bases de dados que já contenham valores ausentes, o sistema faz uma análise de qual algoritmo e qual método de imputação são os mais recomendados e também já fornece a base com os dados imputados. Desta forma, o uso dos benefícios do Aprendizado de Máquina se torna mais acessível, pois usuários leigos podem inserir seus dados no sistema e já receberem informações sobre qual algoritmo é mais recomendado.

1.3 Organização

No Capítulo 2 é apresentada uma revisão bibliográfica sobre Aprendizado de Máquina, dados ausentes e métodos de imputação, assim como a teoria das soluções utilizadas no projeto. Em seguida, no Capítulo 3, são apresentados os módulos que compõem o sistema, assim como a metodologia utilizada e os resultados obtidos. Por fim, no Capítulo 4 encontram-se as conclusões, as considerações para trabalhos futuros e a conexão do projeto com o curso de graduação.

Capítulo 2

REVISÃO BIBLIOGRÁFICA

2.1 Considerações iniciais

Este capítulo tem por objetivo apresentar uma descrição dos conceitos básicos relacionados ao Aprendizado de Máquina, ao problema dos dados ausentes e aos métodos de imputação de dados. Em adição, são apresentados conceitos relacionados à medida de desempenho de classificadores e aos algoritmos estudados para implementar o recomendador proposto nesta monografia.

2.2 Aprendizado de Máquina

Aprendizado de Máquina (AM) é considerado por [Bishop \(2006\)](#) como uma área fundamentada na Estatística e na Inteligência Artificial e que estuda diversos aspectos sobre o processo de aprendizado. Dentre os diversos processos, o de indução é o mais comumente estudado. Ele se baseia na extração de regras e conceitos diretamente dos próprios dados. Apesar do seu intenso uso, nem todas as conclusões deste tipo de processo são sempre verdadeiras ([Mitchell \(1997\)](#)). Dentro deste processo, há duas vertentes básicas que diferenciam o aprendizado, a supervisionada e a não-supervisionada ([Gama et al. \(2011\)](#)).

No aprendizado supervisionado, os dados são previamente rotulados com uma classe e geralmente o objetivo do algoritmo de AM é construir um modelo que separe os dados em classes baseando-se nestes rótulos e conseguir predizer a qual grupo novos dados desconhecidos pertencem. Enquanto no aprendizado não-supervisionado, os dados não são rotulados e comumente o objetivo do algoritmo de AM é descobrir possíveis grupos nos dados, caso existam. Uma outra aplicação desta última vertente é buscar padrões frequentes nos dados.

Os algoritmos de AM utilizam diferentes critérios de preferência para gerar uma hipótese sobre os dados. Estes diferentes critérios são chamados de viés indutivo, e sem ele não há aprendizado ([Mitchell \(1997\)](#)). Isso se dá porque sem viés não há uma restrição no espaço de busca e nem a generalização para classificar novos dados, pois o modelo se tornaria especialista nos dados usados para o treinamento ([von Luxburg e Schoelkopf \(2008\)](#)). Dentre os algoritmos propostos na literatura, alguns se destacam por serem mais difundidos na comunidade, como Árvore de Decisão, *Support Vector Machine* e *Random Forest*, e possuem vieses diferentes.

2.2.1 Árvore de Decisão

Árvores de Decisão pertencem ao grupo de algoritmos que aplicam a estratégia de dividir para conquistar ([Quinlan \(1993\)](#)). Eles particionam recursivamente o conjunto de dados de treinamento até que cada nó folha seja uma das classes do problema. Cada nó não-folha da árvore representa um teste sobre os atributos dos dados e os arcos representam os possíveis valores destes atributos. A escolha do atributo que será utilizado para realizar o teste é feita a partir de medidas de influência deste atributo sobre o conjunto de dados e aquele que possuir maior ganho de informação é escolhido para o teste. Para classificar um novo dado, basta que ele percorra todos os nós da árvore até chegar a um nó folha. A Figura 1 mostra um esquema de Árvore de Decisão para um conjunto de dados que contém dois atributos e três classes.

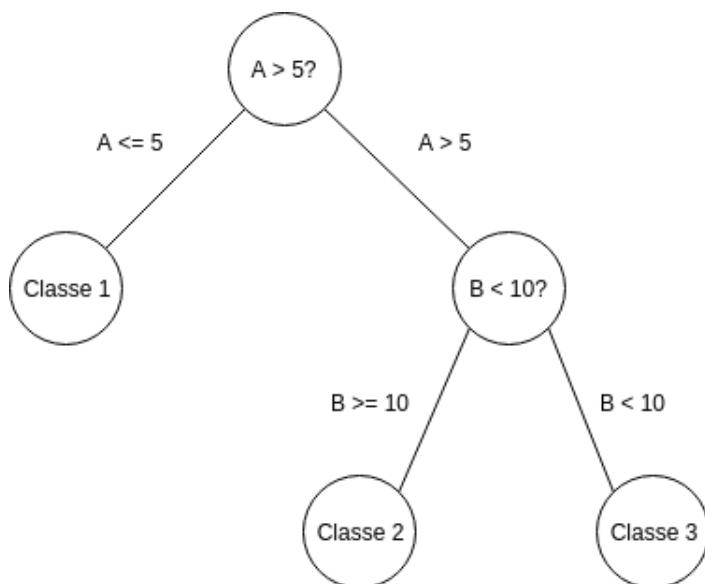


Figura 1 – Exemplo de uma Árvore de Decisão.

Fonte: Elaborada pelo autor.

O algoritmo C4.5 é uma versão de Árvore de Decisão que consegue trabalhar com valores ausentes e valores contínuos, além de podar a árvore em casos em que a taxa de erro da sub-árvore seja elevado quando comparado a utilização de um nó folha ([Quinlan \(1993\)](#), [Ingargiola \(1996\)](#)).

2.2.2 Support Vector Machine

O algoritmo *Support Vector Machine* (SVM) está situado na área de aprendizado de máquina estatístico e a base da sua lógica é encontrar vetores de suporte capazes de encontrar um hiperplano que melhor separe os dados de duas classes distintas. Quanto mais larga a margem de separação entre os exemplos das duas classes, melhor o algoritmo classifica novos dados ([Burges \(1998\)](#)). A Figura 2 mostra graficamente como o SVM divide os dados para um exemplo com duas classes.

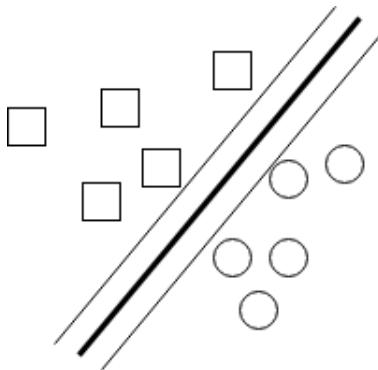


Figura 2 – Exemplo da separação de classes no SVM.

Fonte: Elaborada pelo autor.

Como dificilmente os dados são linearmente separáveis, geralmente é necessário a transformação para um espaço de dimensão maior, a fim de encontrar um hiperplano que seja equidistante das duas classes. Isso é feito através de funções *kernel*. Estas funções podem ser do tipo linear, polinomial, função de base radial e curva sigmoide ([IBM \(2017\)](#)).

2.2.3 Random Forest

O algoritmo *Random Forest* (RF) é um exemplo de algoritmo *ensemble learning*, pois ele gera vários classificadores e agrupa seus resultados. Os classificadores do RF são várias aplicações do algoritmo Árvore de Decisão, cada uma sendo construída com um subconjunto de dados da base original. Além disso, a escolha dos atributos que serão usados para realizar a construção da árvore é feita de maneira aleatória ([Neto et al. \(2015\)](#)). Assim, cada árvore é única, classificando novas entradas de diferentes maneiras. Para obter uma resposta final dada uma nova entrada no modelo gerado pelo algoritmo, todas as árvores a classificam e em seguida o RF seleciona a classe mais frequente de seus elementos preditores, assumindo-a como correta ([Breiman \(2001\)](#)). A Figura 3 ilustra o esquema de árvores do RF.

2.2.4 Validação de Classificadores

Para verificar se o modelo gerado é válido e assertivo, existem várias técnicas que podem ser usadas para isso ([Baranauskas \(2001\)](#)). Dentre estas técnicas, a *k-fold Cross Validation* ganha destaque pelo seu grande uso atualmente. Ela consiste em dividir aleatoriamente o conjunto de dados em k partições de tamanhos iguais e mutuamente exclusivas. Destas k partições, $k - 1$ são usadas para gerar o modelo e a restante é usada para a realização do teste, no qual uma medida de desempenho é aplicada. Este processo se repete k vezes, sempre alterando o conjunto de teste e calculando a medida de desempenho do modelo. No final das k iterações, o desempenho do algoritmo é dado pela média do desempenho de cada iteração. Quando o valor de k é igual à quantidade de amostras do conjunto de dados, este algoritmo passa a ser conhecido como

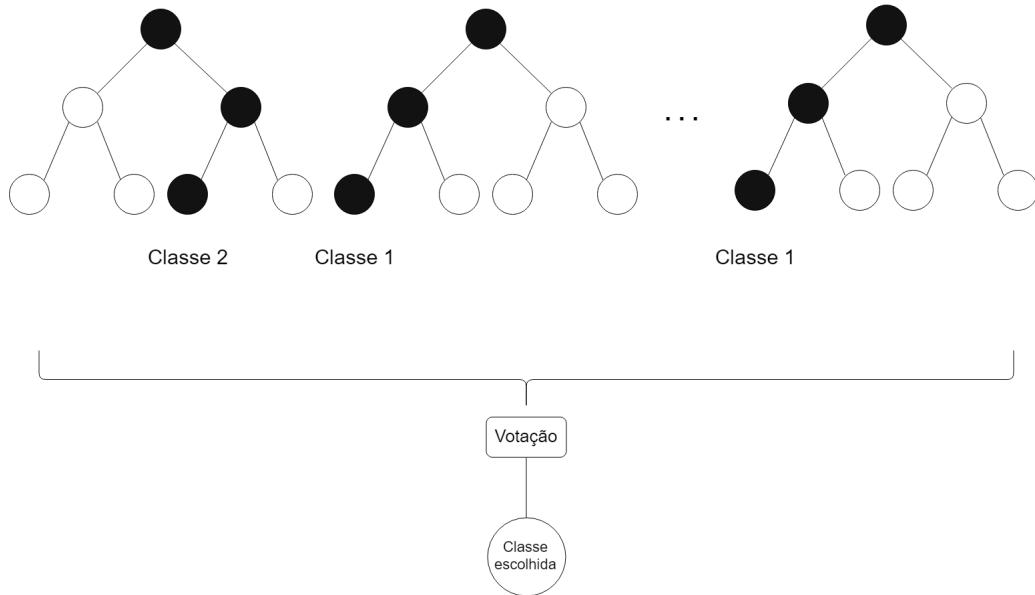


Figura 3 – Exemplo de um esquema do algoritmo *Random Forest*.

Fonte: Elaborada pelo autor.

Leave-one-out Cross Validation. Geralmente este último algoritmo é usado em bases de dados pequenas.

Sobre as medidas de desempenho, podemos destacar a medida de acurácia/erro, que é amplamente utilizada e possui um ótimo desempenho (Ferri, Hernández-Orallo e Modroiu (2009)). A taxa de erro $E(h)$, para uma hipótese h de um conjunto de dados S , de cardinalidade p , é dada pela Equação 2.1, onde y_i é a verdadeira classe do valor $x_i \in S$ e $h(x_i)$ é a classe que o modelo classificou aquela entrada. A função $\delta(y_i, h(x_i))$ vale 0 caso $y_i = h(x_i)$ e vale 1 caso contrário (Souza (2010)).

$$E(h) = \frac{1}{p} \sum_{i=1}^p \delta(y_i, h(x_i)) \quad (2.1)$$

2.3 Dados ausentes

A qualidade dos dados é uma das questões mais importantes na área de Aprendizado de Máquina, em especial para aprendizados baseados em indução, no qual o conhecimento é retirado dos próprios dados (Cios *et al.* (2007)). Se a qualidade destes é baixa, consequentemente o aprendizado não será assertivo, comprometendo a qualidade dos estudos e aplicações que utilizarão o modelo desenvolvido.

Um problema bastante relevante e recorrente é o de dados ausentes, também conhecidos como dados faltantes ou valores desconhecidos (Lobato (2016)). Estes dados ausentes representam uma falta de medição dos valores de um ou mais atributos para alguns dados da base. Esta falta de medição pode ter diversos motivos como defeito nos equipamentos de coleta de dados,

recusa de entrevistados a responder certas perguntas, remoção acidental de valores no banco de dados, entre outros. A Figura 4 ilustra de forma simples um conjunto de dados com valores ausentes.

| Atributos | | | |
|-----------|---|---|---|
| Entradas | | | |
| | ? | | |
| | | | |
| | ? | | |
| | | | |
| | | | |
| | | | |
| | | ? | |
| | ? | | |
| | | | ? |

Figura 4 – Exemplo de um conjunto de dados com valores ausentes.

Fonte: Elaborada pelo autor.

De maneira geral, os motivos dos dados ausentes são processos aleatórios, processos mensuráveis e processos não mensuráveis, e estes geralmente são classificados da seguinte maneira ([Rubin e Little \(2002\)](#) e [Lobato \(2016\)](#)):

- Completamente aleatório (*Missing Completely At Random (MCAR)*): nesta categoria, a probabilidade do atributo ausente é independente dele mesmo e de qualquer outro atributo. Em outras palavras, a sua ausência não tem motivo detectável dentro dos próprios dados.
- Aleatório (*Missing At Random (MAR)*): nesta categoria, a probabilidade do atributo ausente é dependente dos outros atributos dos dados. Ou seja, quando outros atributos possuem certos valores, a probabilidade de um outro atributo não ter medida registrada aumenta.
- Não aleatório (*Not Missing At Random (NMAR)*): nesta categoria, a probabilidade do atributo ausente é dependente do próprio atributo. Isto é, em determinadas situações, dependendo do valor real deste atributo, ele não é coletado.

Como a maioria dos algoritmos de AM não são capazes de trabalhar com dados ausentes no conjunto de entrada, é necessário realizar algum tipo de tratamento sobre eles ([Johansson e Häkkinen \(2006\)](#)). A técnica mais simples para tratá-los é remover todas as entradas do conjunto que possuem algum atributo ausente. Porém, isso gera um desperdício de dados, podendo fazer com que o modelo treinado não seja abrangente o suficiente para classificar de maneira

satisfatória as novas entradas que forem inseridas nele ([Silva \(2010\)](#)). Com isso, a necessidade de um tratamento especial para estes dados ausentes se torna bastante necessária.

2.4 Métodos de imputação

Imputação de dados é uma das técnicas utilizadas para contornar o problema dos dados ausentes. Esta técnica preenche as lacunas de informações estimando os valores a partir do restante dos dados. A característica de precisar apenas dos dados para decidir qual valor inserir onde há dados ausentes é um dos motivos dela ser amplamente utilizada e estudada, pois pode ser aplicada independentemente do algoritmo de AM a ser utilizado posteriormente ([Silva \(2010\)](#)).

As técnicas de imputação são divididas em dois grandes grupos, o da imputação simples (ou imputação única) e o da imputação múltipla. O primeiro visa imputar um único valor para cada dado ausente ([McKnight et al. \(2007\)](#)). Já o segundo consiste em inserir diversos valores nos dados ausentes, testá-los e agregar os diversos resultados obtidos, calculando assim o valor final a ser imputado ([Veroneze \(2011\)](#)).

Diversas técnicas para imputação de dados foram propostas na literatura, sendo que neste projeto são estudadas as técnicas de imputação pela média ou moda, *KNN-Imputation* e *Random Forest*. Em seguida serão apresentados detalhes sobre como estas técnicas funcionam.

2.4.1 Imputação pela Média ou Moda

Esta técnica de imputação simples é a mais comum e geralmente é a forma padrão na maioria das bibliotecas para manipulação de dados ([Brown e Kros \(2003\)](#)). Ela consiste em preencher os valores ausentes de cada atributo do conjunto de dados pela média ou pela moda dos valores medidos nas demais amostras. O primeiro é utilizado para atributos quantitativos e o segundo para atributos qualitativos ([Veroneze \(2011\)](#)).

Este método, além de ser rápido e de fácil implementação, não altera a média geral do atributo na qual é aplicado, mantendo os dados estatísticos da base parecidos com os originais. Porém, há uma perca de variabilidade dos dados, pois os valores extremos não ficam bem representados ([McKnight et al. \(2007\)](#)). Um outro problema desta técnica é que se a distribuição dos valores do atributo em questão não for gaussiana, a média deles não é uma informação muito útil, o que acaba reduzindo a qualidade do modelo que for gerado a partir deles.

2.4.2 KNN Imputation

Neste método de imputação simples, todas as entradas com atributos faltantes são associadas a um subconjunto de entradas no qual mais se assemelham. Para uma dada entrada com um atributo ausente, o algoritmo busca as k entradas mais próximas, chamados k vizinhos mais próximos (do inglês, *k-Nearest Neighbors*), cujo valor deste atributo não é ausente. Encontrado

estas k entradas, a média ou moda do atributo com valor ausente é calculado e imputado ([Silva \(2010\)](#)). Usualmente a distância euclidiana é utilizada para realizar a associação dos k vizinhos com a entrada em questão. A Figura 5 é um exemplo de uma classificação para $k = 3$.

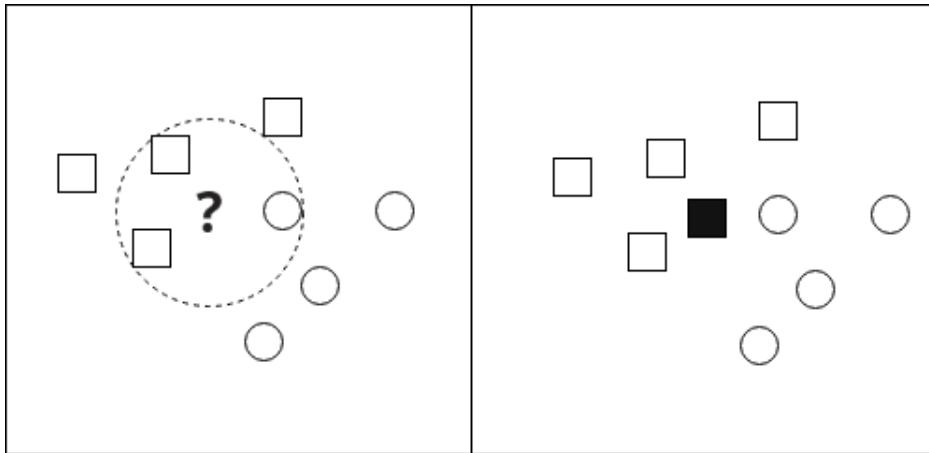


Figura 5 – Exemplo de uma classificação usando KNN para $k = 3$.

Fonte: Elaborada pelo autor.

Um ponto importante desta técnica é a escolha do valor k . Caso k seja muito pequeno, não há amostras o suficiente para associar bem as entradas. Caso k seja muito grande, o risco de associar a amostra a dados não tão parecidos aumenta, o que diminui a acurácia da associação ([Colantonio et al. \(2010\)](#)). O método mais comum para encontrar k é ir incrementando seu valor e ir medindo o desempenho do modelo gerado. Quando a taxa de erro para de diminuir consideravelmente, o valor do k ideal é encontrado.

2.4.3 Random Forest

Este método de imputação múltipla usa o algoritmo RF ([2.2.3](#)) para imputar os dados ausentes. Primeiramente são inseridos dados nos valores ausentes por algum método simples, como valores aleatórios no intervalo das amostras disponíveis ou a média deles. Após isso, é criada uma matriz chamada de Matriz de Proximidade, de ordem $N \times N$, onde N é a quantidade de amostras do conjunto de dados. Ela é inicializada com 1 na sua diagonal principal e 0 em todas as outras posições. Em seguida, é executado o algoritmo RF para classificar os dados. Quando esta execução finaliza, a Matriz de Proximidade é atualizada. Todas as amostras que terminaram num mesmo nó final em uma das árvores do RF têm sua proximidade acrescida em 1. Esta análise é feita para todos os nós finais em todas as árvores geradas. Após a atualização da matriz, seus dados são normalizados pela quantidade de árvores do algoritmo e são usados como pesos em uma média ponderada, com o objetivo de encontrar os novos valores dos atributos que antes estavam ausentes ([Pantanowitz e Marwala \(2009\)](#)). Este processo é repetido até que uma determinada condição seja atingida, como por exemplo um número máximo de repetições.

2.5 Considerações finais

Neste capítulo foram abordados os conceitos básicos de Aprendizado de Máquina, de dados ausentes e da técnica utilizada para amenizar o problema que tal falta de dados acarreta. Alguns algoritmos de AM também foram explicados, assim como alguns dos métodos de imputação existentes.

O próximo capítulo tem como objetivo apresentar o desenvolvimento do trabalho realizado e os resultados obtidos.

Capítulo 3

DESENVOLVIMENTO

3.1 Considerações iniciais

Este capítulo aborda a metodologia utilizada para atingir o objetivo do projeto e a escolha dos métodos de imputação e dos algoritmos de AM utilizados. Os resultados obtidos e as comparações entre os métodos de imputação e os modelos de classificação gerados são apresentados em seguida. Por fim, são relatadas as dificuldades e limitações presentes no trabalho.

3.2 Projeto

A metodologia utilizada para atingir os objetivos propostos na Seção 1.2 se baseou em uma sequência de procedimentos. Primeiramente, foi definido quais seriam os algoritmos de AM estudados, assim como os métodos de imputação que seriam aplicados e testados. Em seguida, foi estudado qual melhor linguagem de programação para usar os algoritmos propostos. Foram analisadas as linguagens de programação *R* e *Python*, pois ambas possuem um grande apelo na comunidade no que diz respeito ao aprendizado de máquina e à análise de dados. *R* foi escolhida pela facilidade no uso e pelo grande número de bibliotecas que são necessárias para este projeto. Por fim, foram criados os módulos que compõem o sistema proposto: um gerador de valores ausentes, um imputador de dados e um analisador de desempenho.

As etapas descritas nesta seção foram desenvolvidas utilizando um computador com processador *Intel i7-6500U*, placa de vídeo *NVIDIA Geforce 940M* e 8GB de memória de acesso aleatório. Os recursos de *software* estão descritos na Tabela 1.

Tabela 1 – Recursos de *software* utilizados.

| Recurso | Categoria | Versão | Plataforma |
|---------|--------------------------|--------|------------|
| Ubuntu | Sistema Operacional (SO) | 16.04 | 64 bits |
| R | Compilador | 3.4.4 | 64 bits |

3.3 Funcionamento do sistema

O sistema funciona de duas formas: no modo de análise e no modo de recomendação. No primeiro, a entrada deve ser uma base de dados completa e, no segundo, uma base que

contenha dados ausentes para que uma recomendação possa ser feita. Os dados devem estar no formato *Attribute-Relation File Format* (ARFF). Esta entrada deve ser inserida no diretório *datasets/original*.

Após a execução dos módulos, as bases imputadas podem ser encontradas no diretório *datasets/imputed* e os gráficos gerados em *plot*. No final da recomendação, uma tabela é exibida contendo o *ranking* das bases imputadas, sendo que a ordem vai do algoritmo de AM e método de imputação que obtiveram o menor erro ao que obtiveram maior erro.

A estrutura do sistema pode ser observada na Figura 6. Mais detalhes da função de cada módulo e de suas entradas e saídas serão descritas na próxima seção.

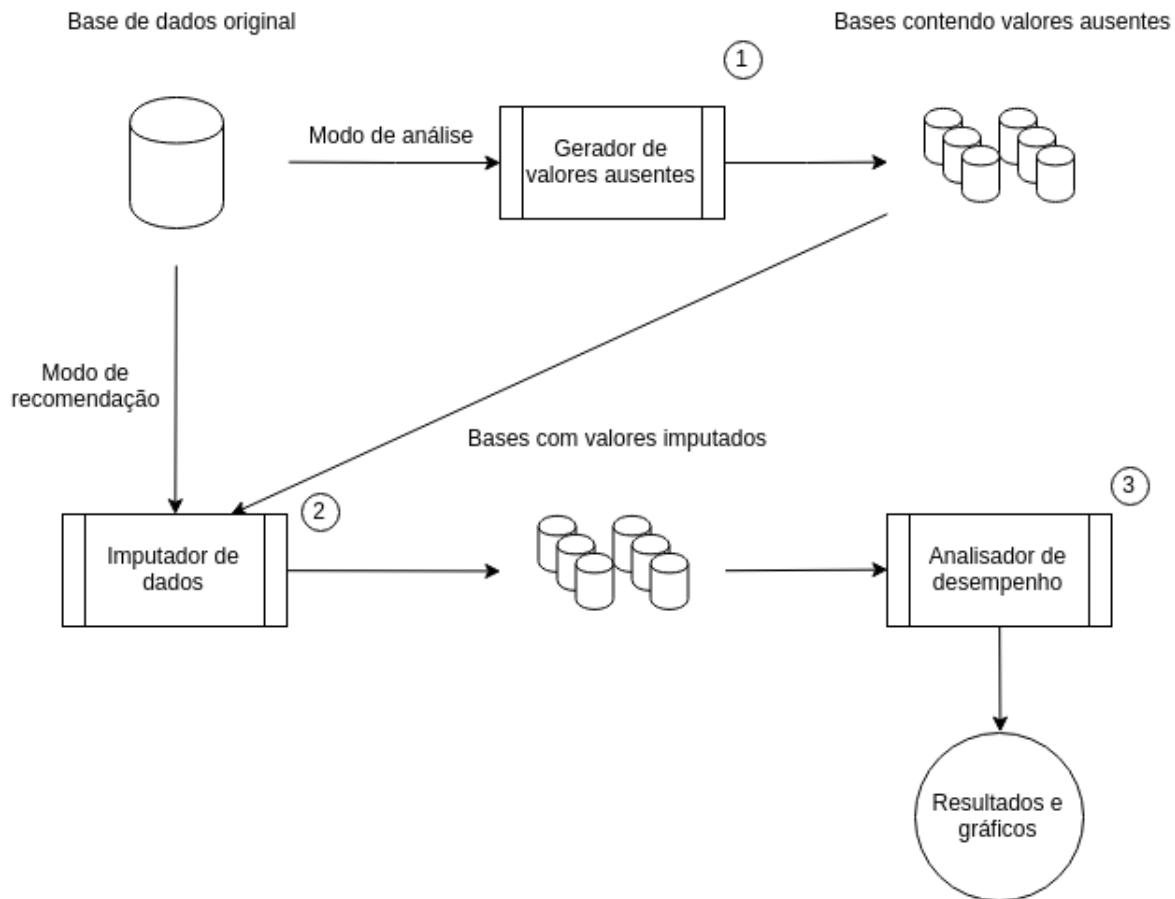


Figura 6 – Estrutura do sistema de desenvolvido.

Fonte: Elaborada pelo autor.

Os códigos desenvolvidos e informações de como utilizá-lo podem ser encontrados no GitHub¹.

¹ <https://github.com/czaghi/recmetimp>

3.4 Descrição das atividades realizadas

Esta seção irá justificar a escolha dos algoritmos de AM e dos métodos de imputação e, em seguida, detalhará os módulos do sistema observados na Figura 6.

3.4.1 Escolha dos algoritmos de AM

A escolha dos algoritmos utilizados nas análises deu-se principalmente aos seus vieses serem distintos. Enquanto a Árvore de Decisão (2.2.1) encontra os atributos com maior entropia nos dados, a fim de fazer comparações com eles e chegar a um resultado mais rapidamente, o SVM (2.2.2) busca distribuir os dados no espaço com o objetivo de separá-los com hiperplanos. Já o algoritmo RF (2.2.3) constrói diferentes Árvores de Decisão baseadas em subconjuntos distintos de dados e de atributos, o que o torna um tanto mais complexo que os outros, porém gera bons resultados.

O uso dos algoritmos pela comunidade também teve peso nesta escolha. Foram escolhidos algoritmos que são amplamente utilizados e difundidos, além de serem implementados em bibliotecas bastante testadas e robustas, o que traz mais confiabilidade no seu uso para o estudo em questão.

3.4.2 Escolha dos métodos de imputação

A escolha dos métodos de imputação foi semelhante a dos algoritmos de AM. Foram estudados diversos métodos propostos na literatura, assim como métodos utilizados pela comunidade. Além disso, todos os métodos escolhidos utilizam a média para determinar qual o valor do dado ausente, mas de maneiras distintas. O algoritmo de média descrito em (2.4.1) utiliza a média dos valores presentes para aquele atributo em todo o conjunto de dados. O KNN-Imputation (2.4.2) seleciona a média somente dos k dados que mais se assemelham a ele. Já o RF (2.4.3) calcula a média baseado na semelhança da classificação das diversas Árvores de Decisão que o compõem. Desta forma, obtém-se diversos resultados para uma mesma base de dados, sendo que cada um possui um desempenho distinto dependendo das características dos dados e de como os valores ausentes se distribuem no meio desses.

3.4.3 Gerador de valores ausentes

Para realizar a análise da eficiência dos métodos de imputação neste projeto, foi necessário construir um gerador de valores ausentes, a fim de poder medir o desempenho dos modelos gerados pelos algoritmos de AM a partir das bases imputadas em relação ao dos modelos gerados utilizando a base de dados original. Este módulo está representado em 1 na Figura 6, e é utilizado apenas para o modo de análise, pois no modo de recomendação a base original já contém valores ausentes.

Para construir este gerador, utilizou-se a biblioteca *missForest*, pois ela possui um método que gera valores aleatórios em um conjunto de dados a uma porcentagem escolhida. Desta forma, é possível variar a porcentagem de dados ausentes que se deseja ter no conjunto de dados, variando o estudo a ser feito conforme desejado. No gerador foram fixadas as porcentagens de 5%, 15% e 25% de valores ausentes.

Além disso, o módulo gera 10 bases de dados distintas para cada porcentagem quando é executado. Isso é feito devido aos dados ausentes serem do tipo MCAR. Como não há controle de quais atributos terão mais dados ausentes e nem de como estes dados estão distribuídos pela base, essa diversificação de bases de dados é extremamente importante para realizar o estudo. Dependendo desta distribuição de dados, os métodos de imputação encontrarão valores distintos a serem imputados e isso afeta diretamente os algoritmos de AM que serão executados em seguida. Com uma diversidade de bases de dados, uma análise mais assertiva pode ser feita posteriormente.

3.4.4 Imputador de dados

Após a geração das bases de dados com valores ausentes no modo de análise, ou com a inserção da base de dados no modo de recomendação, o próximo passo é aplicar os métodos de imputação nestas bases. As técnicas escolhidas, os parâmetros e as bibliotecas utilizadas para aplicá-las estão listadas na Tabela 2.

Tabela 2 – Técnicas de imputação utilizadas.

| Técnica | Parâmetros | Biblioteca |
|-----------------------|--|--------------|
| Imputação pela média | <i>default</i> | missForest |
| <i>KNN-Imputation</i> | $k = \{3, 5, 9\}$, <i>imp_var</i> = FALSE | VIM |
| <i>Random Forest</i> | <i>default</i> | randomForest |

O módulo, que é representado por 2 na Figura 6, faz uma varredura no diretório que contém os dados ausentes, carregando uma base de dados por vez na memória. Com a base carregada, os métodos de imputação citados acima são aplicados a ela, resultando em novas 5 bases de dados. Cada uma destas bases agora possuem todos os dados preenchidos, além de um novo atributo, que serve para indicar qual foi o método utilizado para realizar a imputação. Sendo assim, no modo de análise, obtém-se 50 bases de dados distintas para realizar o estudo dos métodos de imputação e dos algoritmos de AM, todas geradas a partir de uma única base de dados original.

3.4.5 Analisador de desempenho

Neste módulo, representado por 3 na Figura 6, as bases de dados imputadas e a original, esta última apenas no modo de análise, são testadas uma a uma nos algoritmos de AM. Com o objetivo de construir um módulo escalável e de fácil manutenção, foi utilizado o *framework*

Machine Learning in R (MLR). Este *framework* possui uma interface simples e que permite adicionar vários algoritmos de AM de uma maneira bastante prática, o que tornou esse objetivo possível.

Para construir modelos de classificação, foram utilizados os algoritmos e as bibliotecas elencados na Tabela 3. Todos os algoritmos foram utilizados com seus parâmetros padrões da biblioteca.

Tabela 3 – Algoritmos de AM utilizados.

| Algoritmo | Biblioteca |
|-----------|--------------|
| C4.5 | RWeka |
| SVM | e1071 |
| RF | randomForest |

Para medir o desempenho dos modelos gerados, foi utilizado o algoritmo *k-fold Cross Validation* (2.2.4), com $k = 10$. Do modelo final, extraiu-se porcentagem de erro das classificações do modelo, descrita pela Equação 2.1. Para cada base de dados foram gerados 30 modelos para cada algoritmo de AM, e a média do erro destes modelos foi usada como valor final para a base de dados, o método de imputação e o algoritmo em questão.

Calculada a taxa de erro para cada um dos AM, obtém-se um conjunto de dados composto por um identificador da base, o método de imputação que foi utilizado nela (ou nenhum, no caso da base de dados original), o algoritmo de AM e o erro do algoritmo. Com estes dados em mãos, é possível compará-los e fazer uma recomendação de qual método de imputação e algoritmo são os mais recomendados para o caso em questão. Além disso, este módulo gera gráficos destes resultados. No caso do modo de análise, para cada algoritmo de AM e porcentagem de dados ausentes, é gerado um *Heat Map* dos métodos de imputação \times bases de dados. Já no caso do modo de recomendação, apenas um *Heat Map* é gerado, sendo dos métodos de imputação \times algoritmos de AM.

3.5 Resultados obtidos

Inicialmente serão apresentados os resultados para a execução do sistema no modo de análise. Desta forma, espera-se mostrar a necessidade de escolher um bom algoritmo de AM para cada base de dados, assim como um método de imputação adequado. Em seguida, são apresentados os resultados para duas bases de dados contendo valores ausentes no modo de recomendação.

Todas as bases utilizadas podem ser encontradas no projeto OpenML ([Vanschoren et al. \(2013\)](#)). Foram utilizadas as bases originais no modo de análise e bases modificadas no modo de recomendação, onde alguns dados foram removidos de maneira aleatória para a realização do experimento.

3.5.1 Resultados para o modo de análise

Foram testadas 10 bases de dados neste modo, que estão descritas na Tabela 4 juntamente com algumas de suas características.

Tabela 4 – Bases de dados utilizadas na extração dos dados em modo de análise.
(*num.*: numérico; *cat.*: categórico)

| Bases de Dados | Amostras | Atributos | Classes |
|--------------------------|----------|-----------------|---------|
| <i>Cont. Met. Choice</i> | 1473 | 2 num. e 7 cat. | 3 |
| <i>Diabetes</i> | 768 | 8 num. | 2 |
| <i>Ionosphere</i> | 351 | 33 num. | 2 |
| <i>Iris</i> | 150 | 4 num. | 3 |
| <i>LED</i> | 500 | 7 num. | 10 |
| <i>Monk's problems 1</i> | 556 | 6 cat. | 2 |
| <i>Parkinsons</i> | 195 | 22 num. | 2 |
| <i>PC Req</i> | 320 | 7 num. e 1 cat. | 2 |
| <i>TAE</i> | 151 | 3 num. e 2 cat. | 3 |
| <i>Wine</i> | 178 | 13 num. | 3 |

A análise para as bases *Wine* e *Contraceptive Method Choice* são discutidas a seguir. Para as demais bases, os resultados podem ser encontrados no Apêndice A.

3.5.1.1 Análise da base Wine

Na Tabela 5 encontram-se os valores dos erros obtidos pelos modelos de AM para os dados originais. Nas Figuras 7, 8 e 9 estão representados os dados obtidos pelo sistema após as análises e estruturados em *Heat Maps*. Neles, cores mais frias (tons de azul) representam um erro menor do modelo e cores mais quentes (tons de vermelho) representam um erro maior.

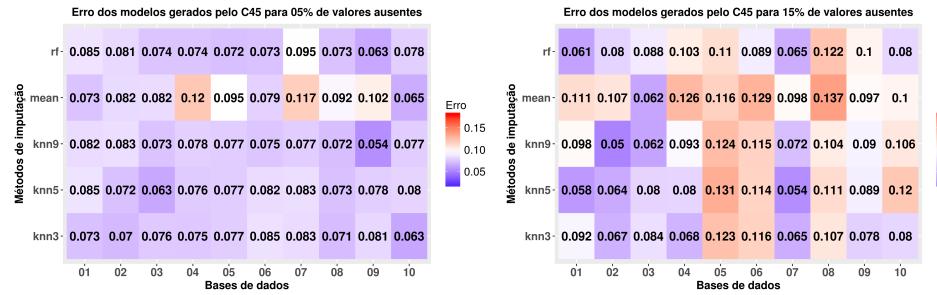
Tabela 5 – Erros obtidos para a base de dados *Wine*.

| Algoritmo de AM | Erro |
|-----------------|-------------------|
| C4.5 | $0,068 \pm 0,015$ |
| SVM | $0,017 \pm 0,003$ |
| RF | $0,019 \pm 0,003$ |

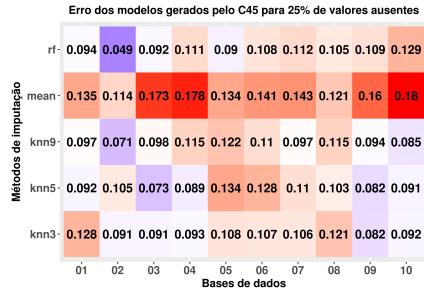
Pela análise dos gráficos, nota-se que o desempenho do SVM e do RF são muito parecidos e que ambos são superiores ao C4.5, que apresentou um resultado não satisfatório para esta base. Nota-se também a grande influência da porcentagem de dados ausentes no desempenho dos modelos gerados, pois conforme esta porcentagem aumenta, a porcentagem média de erros dos modelos gerados pelos AM aumenta.

Comparando-se os erros das bases imputadas com os da base original, nota-se que para uma baixa porcentagem de dados ausentes o desempenho dos algoritmos são satisfatórios, sendo bem próximos ao erro da base original. Porém, pelos dados não é possível dizer qual método de imputação se destaca dos demais, pois seu desempenho varia de caso a caso.

Figura 7 – Resultados do AM C4.5 para a base Wine.

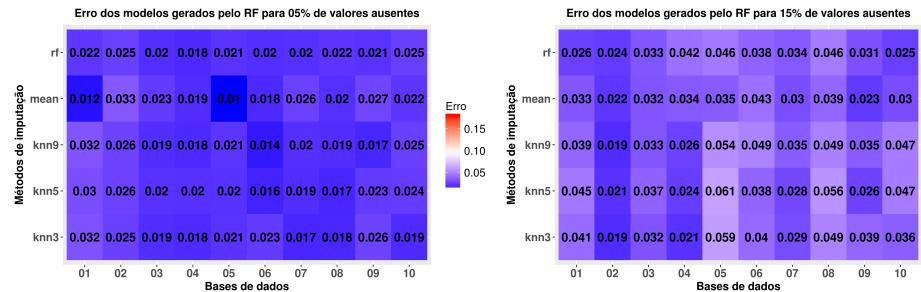


(a) Análise para 5% de dados ausentes. (b) Análise para 15% de dados ausentes.

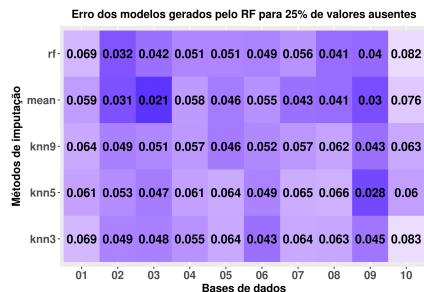


(c) Análise para 25% de dados ausentes.

Figura 8 – Resultados do AM RF para a base Wine.



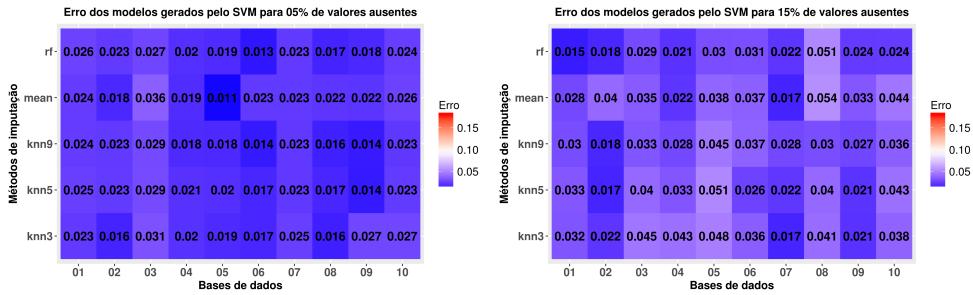
(a) Análise para 5% de dados ausentes. (b) Análise para 15% de dados ausentes.



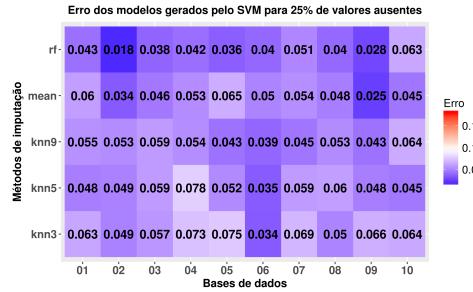
(c) Análise para 25% de dados ausentes.

3.5.1.2 Análise da base Contraceptive Method Choice

Na Tabela 6 estão listados os valores dos erros obtidos para a base de dados original nos modelos gerados. Nas Figuras 10, 11 e 12 encontram-se os resultados obtidos após a execução dos módulos do sistema, na mesma estrutura de Heat Maps da base anterior.

Figura 9 – Resultados do AM SVM para a base *Wine*.

(a) Análise para 5% de dados ausentes. (b) Análise para 15% de dados ausentes.



(c) Análise para 25% de dados ausentes.

Tabela 6 – Erros obtidos para a base de dados *Contraceptive Method Choice*.

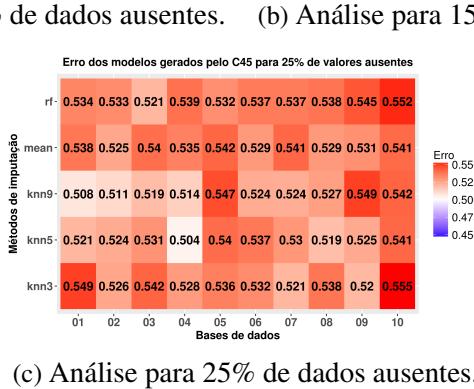
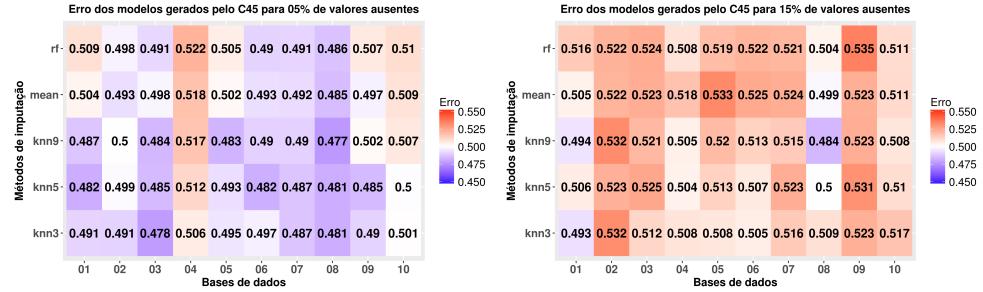
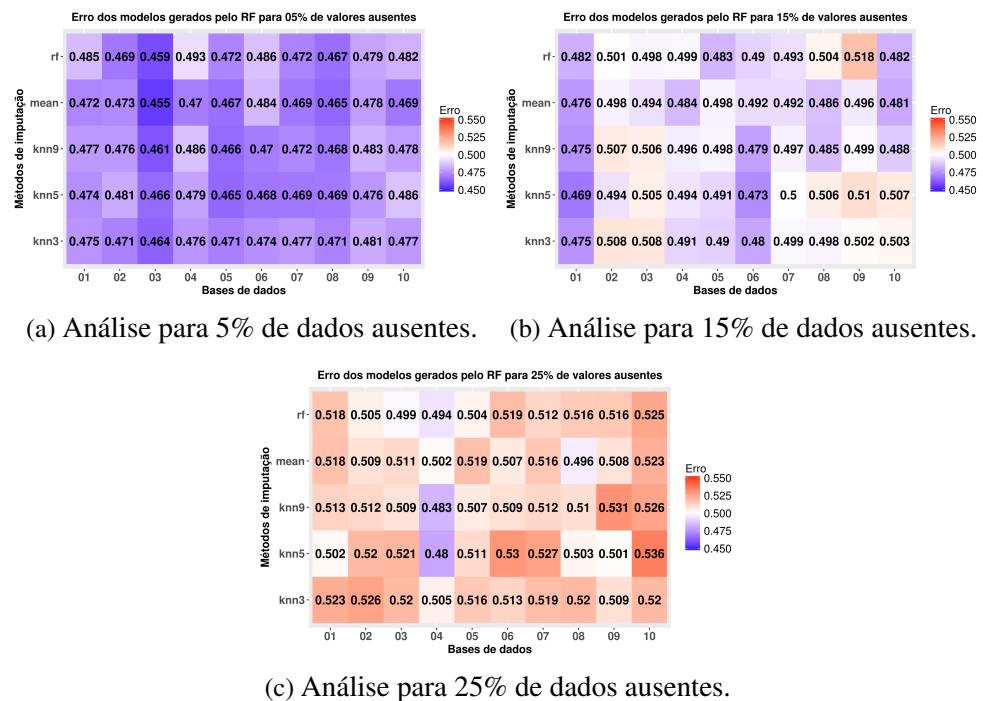
| Algoritmo de AM | Erro |
|-----------------|---------------|
| C4.5 | 0,487 ± 0,007 |
| SVM | 0,447 ± 0,004 |
| RF | 0,461 ± 0,006 |

Analizando graficamente os resultados obtidos, é possível detectar qual é o AM mais recomendado para a base em questão, sendo este o SVM. Seu desempenho é levemente superior ao RF e bem melhor que os resultados obtidos pelo C4.5, que obteve o pior desempenho. Da mesma forma que no caso anterior, nota-se também influência da porcentagem de dados ausentes no desempenho dos modelos gerados.

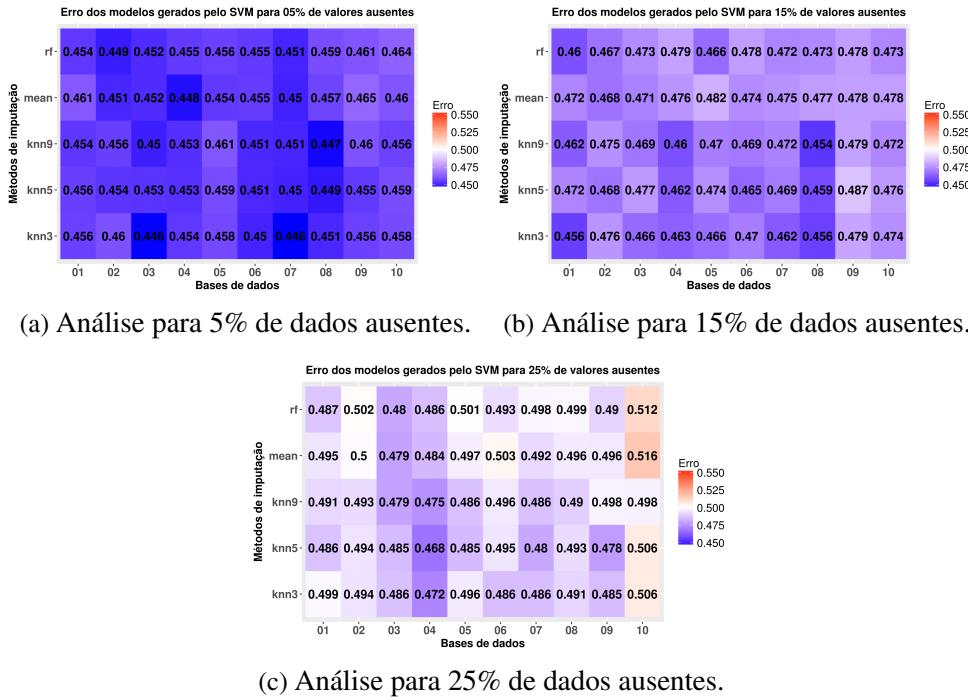
Neste caso também é possível analisar que, para uma baixa porcentagem de dados ausentes, as bases imputadas possuem um desempenho muito parecido com o da base original.

Pelos resultados obtidos nas duas bases acima, juntamente com as descritas no Apêndice A, não é possível fazer uma análise assertiva a respeito de qual método de imputação se destaca dos demais, pois para cada caso um método obteve um resultado melhor que o outro. Estratégias que possibilitam esta análise e que não foram utilizadas neste projeto são discutidas e justificadas na Seção 3.6. Sendo assim, a recomendação feita neste projeto é baseada na análise de todos os métodos e todos os algoritmos para a base inserida.

No entanto, com estas amostras é possível notar que existe a necessidade de escolher um algoritmo de AM para cada caso, pois não há um algoritmo que se destaque para todas as bases. O mesmo pode ser dito sobre os métodos de imputação: eles variam muito dentro de uma

Figura 10 – Resultados do AM C4.5 para a base *Contraceptive Method Choice*.Figura 11 – Resultados do AM RF para a base *Contraceptive Method Choice*.

mesma base, para um mesmo AM, e isso pode ser observado em vários casos, como na Base 2 da Figura 7c, na Base 8 da Figura 10b e na Base 9 da Figura 11c.

Figura 12 – Resultados do AM SVM para a base *Contraceptive Method Choice*.

3.5.2 Resultados para o modo de recomendação

As duas bases de dados utilizadas para exemplificar os resultados obtidos para a recomendação feita pelo sistema são as *Molecular Biology (Promoter Gene Sequences)* e *Vehicle*. As características destas bases estão descritas na Tabela 7.

Tabela 7 – Bases de dados utilizadas no modo de recomendação.
(*num.*: numérico; *cat.*: categórico)

| Bases de dados | Amostras | Atributos | Classes | Valores ausentes (%) |
|--------------------------|----------|-----------|---------|----------------------|
| <i>Molecular Biology</i> | 106 | 58 cat. | 2 | 15 |
| <i>Vehicle</i> | 846 | 18 num. | 4 | 10 |

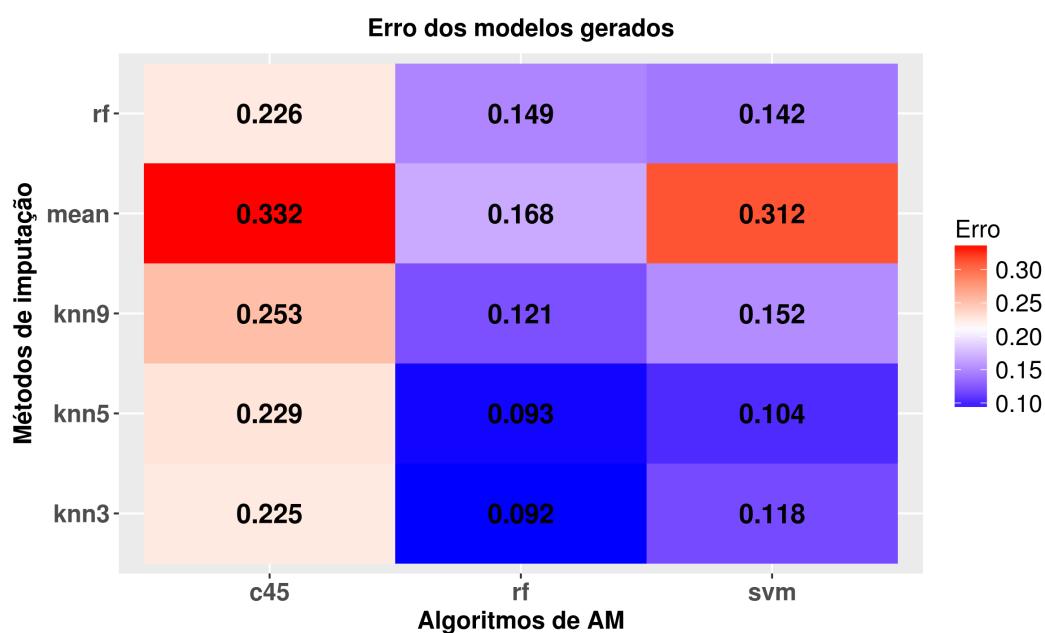
3.5.2.1 Recomendações para a base *Molecular Biology (Promoter Gene Sequences)*

Quando o sistema é executado neste modo, ele gera uma tabela, que pode ser observada na Tabela 8, além de uma análise gráfica, que pode ser observado na Figura 13.

Para esta base, observa-se uma pequena vantagem do uso do algoritmo RF em relação ao SVM, porém uma grande vantagem em relação ao C4.5. Nota-se também que o método de imputação KNN, para os valores de $k = 3, 5$, se destaca dos demais métodos. Com este resultado, dependendo da escolha feita pelo usuário o desempenho do modelo gerado, pode variar numa faixa de mais de 20% de erro, o que é bastante crítico.

Tabela 8 – Resultado da recomendação para a base *Molecular Biology (Promoter Gene Sequences)*.

| Método de Imputação | Algoritmo de AM | Erro médio dos modelos |
|----------------------------|------------------------|-------------------------------|
| KNN3 | RF | 0.092 ± 0.013 |
| KNN5 | RF | 0.093 ± 0.016 |
| KNN5 | SVM | 0.104 ± 0.013 |
| KNN3 | SVM | 0.118 ± 0.013 |
| KNN9 | RF | 0.121 ± 0.016 |
| RF | SVM | 0.142 ± 0.002 |
| RF | RF | 0.149 ± 0.005 |
| KNN9 | SVM | 0.152 ± 0.011 |
| Mean | RF | 0.168 ± 0.013 |
| KNN3 | C4.5 | 0.225 ± 0.025 |
| RF | C4.5 | 0.226 ± 0.036 |
| KNN5 | C4.5 | 0.229 ± 0.027 |
| KNN9 | C4.5 | 0.253 ± 0.021 |
| Mean | SVM | 0.314 ± 0.042 |
| Mean | C4.5 | 0.332 ± 0.025 |

Figura 13 – Resultado gráfico das análises feitas com a base *Molecular Biology (Promoter Gene Sequences)* no modo de recomendação.

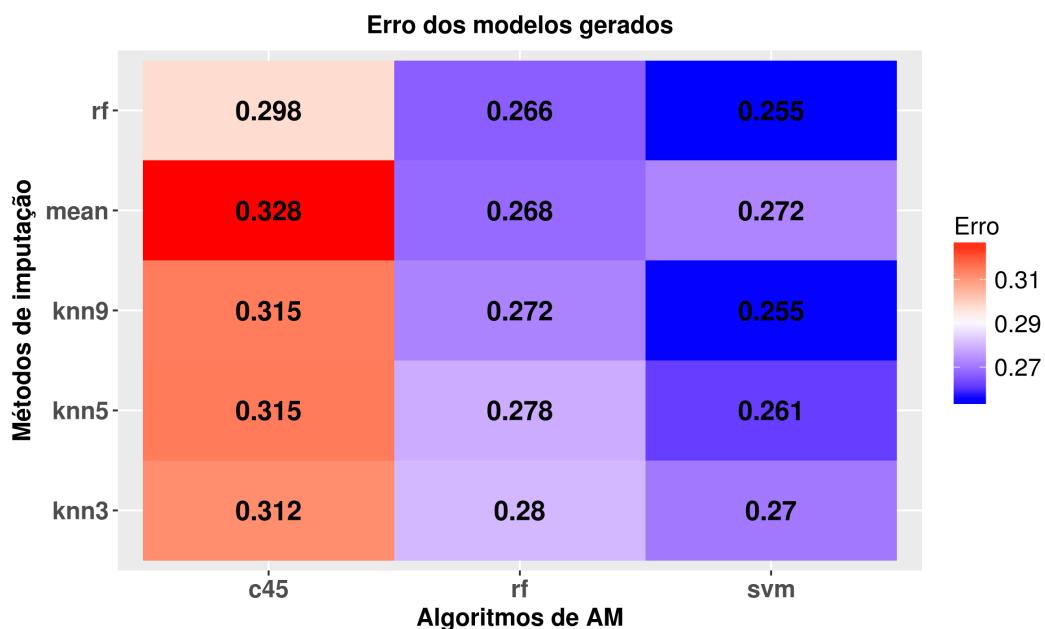
3.5.2.2 Recomendações para a base Vehicle

Os resultados gerados pelo sistema para esta base podem ser observados na Tabela 9 e na Figura 14.

Neste caso, nota-se que o SVM é o AM mais adequado para esta base de dados. Pode-se observar também que não há um método de imputação que se destaca em todos os algoritmos, sendo que o RF ou o KNN com $k = 9$ são os recomendados a se usar nesta base. A escolha do método de imputação e de algoritmo de AM pode fazer com que o erro médio do modelo gerado

Tabela 9 – Resultado da recomendação para a base *Vehicle*.

| Método de Imputação | Algoritmo de AM | Erro médio dos modelos |
|---------------------|-----------------|------------------------|
| RF | SVM | 0.255 ± 0.008 |
| KNN9 | SVM | 0.255 ± 0.006 |
| KNN5 | SVM | 0.261 ± 0.007 |
| RF | RF | 0.265 ± 0.007 |
| Mean | RF | 0.268 ± 0.009 |
| KNN3 | SVM | 0.269 ± 0.008 |
| KNN9 | RF | 0.272 ± 0.007 |
| Mean | SVM | 0.272 ± 0.007 |
| KNN5 | RF | 0.278 ± 0.008 |
| KNN3 | RF | 0.279 ± 0.007 |
| RF | C4.5 | 0.298 ± 0.009 |
| KNN3 | C4.5 | 0.312 ± 0.009 |
| KNN9 | C4.5 | 0.315 ± 0.009 |
| KNN5 | C4.5 | 0.315 ± 0.013 |
| Mean | C4.5 | 0.327 ± 0.014 |

Figura 14 – Resultado gráfico das análises feitas com a base *Vehicle* no modo de recomendação

varie cerca de 7%.

3.6 Dificuldades e limitações

A maior limitação neste trabalho foi a dificuldade de extrair meta-dados para realizar análises mais assertivas sobre situações em que um algoritmo de AM seria mais recomendado, assim como qual método é o mais recomendado para cada caso. Originalmente tinha-se a ideia de extrair meta-dados e gerar meta-modelos de recomendação de métodos de imputação e de

algoritmos de AM, e para isso uma grande quantidade de dados é necessária. Sendo assim, o número de métodos e algoritmos analisados teve que ser reduzido, pois o poder computacional necessário para extrair uma quantidade de dados considerável é alto.

Entretanto, os métodos de extração de meta-dados mais avançados não levam os dados ausentes em consideração, algo que é fundamental para o estudo em questão. Desta forma, foi estudada a possibilidade de criar um modelo de extração com características gerais sobre os dados, porém os resultados foram muito abrangentes e não conclusivos. Como o desenvolvimento de técnicas de extração de meta-dados que incluíssem informações de dados ausentes seria bastante complexo, elas não foram desenvolvidas, optando-se por fazer a recomendação dos algoritmos e técnicas testando-se todas as combinações na base de dados inserida.

3.7 Considerações finais

Neste capítulo foi descrito como foram feitas as escolhas dos métodos de imputação e dos algoritmos de AM que são recomendados de acordo com seus desempenhos nas bases de dados testadas. Uma descrição de como o projeto funciona, assim como os módulos que o compõe, foi dada em seguida. Após isso, os resultados obtidos, tanto para o modo de análise quanto para o modo de recomendação, foram apresentados. Por fim, foram apontadas as dificuldades e limitações encontradas no desenvolvimento do projeto. O próximo capítulo apresenta as conclusões obtidas.

Capítulo 4

CONCLUSÃO

4.1 Contribuições

Este projeto tem como objetivo montar um sistema de recomendação e de análise de algoritmos de AM e de métodos de imputação para bases contendo valores ausentes. Para isso, partindo de uma base de dados completa, foram criadas várias bases contendo valores ausentes em três porcentagens diferentes. Após isso, os métodos de imputação foram aplicados a estas bases e o desempenho dos modelos gerados pelos algoritmos de AM foram medidos e comparados. Notou-se que, para baixas porcentagens de valores ausentes, os métodos de imputação fornecem um resultado muito bom. Para bases com uma grande porcentagem de dados ausentes, o desempenho cai, porém não de uma forma abrupta como seria caso as entradas contendo valores ausentes fossem removidas do conjunto de dados.

Também foi possível notar que cada algoritmo de AM se destaca para algumas bases, entretanto não tem um desempenho satisfatório para outras. Os métodos de imputação se comportam da mesma forma. Desse modo, é cabível a construção de um sistema de recomendação para eles.

4.2 Relacionamento entre o Curso e o Projeto

O desenvolvimento deste projeto só foi possível graças ao conhecimento adquirido em diversas disciplinas oferecidas pelo curso de Engenharia de Computação. Em especial, a disciplina de Inteligência Artificial em que foram ensinados a base do aprendizado de máquina, os tipos, os paradigmas e noções gerais, sendo de grande ajuda no desenvolvimento. Outra disciplina que se relaciona diretamente com o projeto é Avaliação de Desempenho de Sistemas Computacionais, que auxiliou bastante na realização as análises dos resultados obtidos e as comparações dos algoritmos. Para gerenciar e estruturar o projeto, os conteúdos aprendidos na disciplina de Engenharia de Software foram de grande ajuda. Contudo, sem as disciplinas básicas de computação, nada disso seria possível. Os conteúdos vistos nas disciplinas Introdução à Ciência da Computação I e II e Algoritmos de Estrutura de Dados I e II foram essenciais para o entendimento das outras disciplinas, assim como para o desenvolvimento dos códigos deste projeto.

4.3 Considerações sobre o Curso de Graduação

O curso de Engenharia de Computação da Escola de Engenharia de São Carlos (EESC) e Instituto de Ciências e de Computação (ICMC) possui uma grade curricular bastante ampla e versátil, abrangendo tópicos tanto do curso de Ciências de Computação quanto de Engenharia Elétrica. Esta versatilidade é muito boa, pois o aluno tem uma base de conhecimento teórico bastante sólida em ambas as áreas, porém isso não era devidamente aproveitado. Quase nenhuma disciplina da vertente de Ciências da Computação tinha interação com as da Engenharia Elétrica e vice-versa. Isso dava a sensação de estar cursando dois cursos diferentes ao mesmo tempo e a frequente troca de contexto é algo que sempre foi difícil no curso.

Além disso, a carga horária do curso é bastante pesada, ainda mais quando muitos professores não respeitam os créditos da disciplina. É clara a extração dos créditos de trabalho em diversas disciplinas, sendo que até mesmo disciplinas que não possuem créditos de trabalho tiveram trabalhos bastante extensos que valiam uma porcentagem considerável da nota final. Isso obrigava os alunos a darem mais atenção a algumas disciplinas do que outras, o que é extremamente errado num curso de graduação, pois todas que constam na grade são a base do conhecimento para o futuro do aluno.

Um outro ponto de mudança é a obrigatoriedade de algumas disciplinas do curso que poderiam ser optativas, pois ou possuem uma alta relação com alguma ênfase oferecida no curso ou juntas poderiam formar uma nova ênfase. Nos últimos anos do curso são ministradas disciplinas obrigatórias que são eletivas no curso de Ciências da Computação e no de Engenharia Elétrica. Como nesta etapa o aluno já está maduro e apto a escolher a ênfase que quer seguir, estas disciplinas poderiam ser removidas da grade obrigatória, aliviando a alta carga do curso e deixando mais tempo para os alunos focarem no que realmente se interessam e possuem talento.

4.4 Trabalhos Futuros

Para trabalhos futuros, seria interessante incrementar o sistema inserindo com mais métodos de imputação a serem analisados, assim como mais algoritmos de AM. Para algumas bases testadas, os algoritmos C4.5, RF e SVM não são bons, o que limita o uso do sistema. Este projeto foi desenvolvido de maneira que torna esta expansão simples de ser aplicada.

Podem ser desenvolvidos métodos de extração de meta-dados confiáveis sobre os resultados das análises, e com eles, gerar meta-modelos para recomendar tanto o melhor algoritmo de AM quanto o método de imputação, sem a necessidade de testar a base em todos como é feito atualmente. Com isso, o tempo que leva para recomendar uma nova base diminuiria consideravelmente, pois a tarefa seria apenas extrair os meta-dados da nova base e classificá-los no meta-modelo.

Outro trabalho interessante que pode ser desenvolvido para complementar o que foi feito

é adicionar um módulo que testa técnicas de remoção de *outliers* e também recomenda a que forneceu o melhor desempenho.

REFERÊNCIAS

- BARANAUSKAS, J. A. **Extração automática de conhecimento por múltiplos indutores.** Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2001. Citado na página 7.
- BATISTA, G. E. A. P. A. **Pré-processamento de dados em aprendizado de máquina supervisionado.** Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2003. Citado na página 1.
- BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics).** [S.l.]: Springer-Verlag New York, Inc., 2006. Citado 2 vezes nas páginas 1 e 5.
- BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, Oct 2001. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1023/A:1010933404324>>. Citado na página 7.
- BROWN, M. L.; KROS, J. F. Data mining and the impact of missing data. **Industrial Management & Data Systems**, v. 103, n. 8, p. 611–621, 2003. Disponível em: <<https://doi.org/10.1108/02635570310497657>>. Citado na página 10.
- BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. **Data Mining and Knowledge Discovery**, v. 2, p. 121–167, 1998. Citado na página 6.
- CIOS, K. J.; PEDRYCZ, W.; SWINIARSKI, R. W.; KURGAN, L. A. **Data Mining: A Knowledge Discovery Approach.** [S.l.]: Springer-Verlag New York, Inc., 2007. Citado na página 8.
- COLANTONIO, A.; PIETRO, R. D.; OCELLO, A.; VERDE, N. V. Abba: adaptive bicluster-based approach to impute missing values in binary matrices. In: **SAC**. [S.l.: s.n.], 2010. Citado na página 11.
- DONCEL, L. **A Era do algoritmo chegou e seus dados são um tesouro.** 2018. <https://brasil.elpais.com/brasil/2018/03/01/economia/1519921981_137226.html>. Acessado em: 2018-03-15. Citado na página 1.
- FERRI, C.; HERNÁNDEZ-ORALLO, J.; MODROIU, R. An experimental comparison of performance measures for classification. **Pattern Recogn. Lett.**, Elsevier Science Inc., New York, NY, USA, v. 30, n. 1, p. 27–38, jan. 2009. ISSN 0167-8655. Disponível em: <<http://dx.doi.org/10.1016/j.patrec.2008.08.010>>. Citado na página 8.
- GAMA, J.; CARVALHO, A. C. P. L. F.; FACELI, K.; LORENA, A. C. **Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina.** [S.l.]: LTC, 2011. Citado na página 5.
- IBM. **IBM Knowledge Center.** 2017. <https://www.ibm.com/support/knowledgecenter/pt-br/SS3RA7_17.1.0/modeler_mainhelp_client_ddita/clementine/svm_howwork.html>. Acessado em: 2018-03-31. Citado na página 7.
- INGARGIOLA, G. **Building Classification Models: ID3 and C4.5.** 1996. <<https://cis.temple.edu/~ingargio/cis587/readings/id3-c45.html>>. Citado na página 6.

JOHANSSON, P.; HäKKINEN, J. Improving missing value imputation of microarray data by using spot quality weights. **BMC Bioinformatics**, 2006. Citado na página 9.

LESOT, M.-J.; D'ALLONNES, A. R. Credit-card fraud profiling using a hybrid incremental clustering methodology. In: HÜLLERMEIER, E.; LINK, S.; FOBER, T.; SEEGER, B. (Ed.). **Scalable Uncertainty Management**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 325–336. ISBN 978-3-642-33362-0. Citado na página 1.

LOBATO, F. M. F. **Estratégias evolucionárias para otimização no tratamento de dados ausentes por imputação múltipla de dados**. Tese (Doutorado) — Instituto de Tecnologia, Universidade Federal do Pará, 2016. Citado 2 vezes nas páginas 8 e 9.

MCKNIGHT, P. E.; MCKNIGHT, K. M.; SIDANI, S.; FIGUEREDO, A. J. **Missing Data: A Gentle Introduction**. [S.l.]: The Guilford Press, 2007. Citado na página 10.

MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997. Citado 2 vezes nas páginas 2 e 5.

NETO, C. G.; FONSECA, L. M. G.; KÖRTING, T.; SANCHES, I.; EBERHARDT, I.; BENDINI, H.; MARUJO, R.; TRANBAQUINI, K. Classificação automática de áreas cafeeiras utilizando imagens de sensoriamento remoto e técnicas de mineração de dados. 04 2015. Citado na página 7.

PANTANOWITZ, A.; MARWALA, T. Missing data imputation through the use of the random forest algorithm. In: YU, W.; SANCHEZ, E. N. (Ed.). **Advances in Computational Intelligence**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 53–62. ISBN 978-3-642-03156-4. Citado na página 11.

PORTUGAL, I.; ALENCAR, P. S. C.; COWAN, D. D. The use of machine learning algorithms in recommender systems: A systematic review. **CoRR**, abs/1511.05263, 2015. Disponível em: <<http://arxiv.org/abs/1511.05263>>. Citado na página 1.

QUINLAN, J. R. **C4.5: programs for machine learning**. [S.l.]: Morgan Kaufmann Publishers Inc., 1993. Citado na página 6.

RUBIN, D. B.; LITTLE, R. J. A. **Statistical Analysis with Missing Data**. [S.l.]: John Wiley & Sons, Inc., 2002. Citado na página 9.

SILVA, J. de A. **Substituição de valores ausentes: uma abordagem baseada em um algoritmo evolutivo para agrupamento de dados**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2010. Citado 3 vezes nas páginas 1, 10 e 11.

SOUZA, B. F. de. **Meta-aprendizagem aplicada à classificação de dados de expressão gênica**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2010. Citado 2 vezes nas páginas 1 e 8.

VANSCHOREN, J.; RIJN, J. N. van; BISCHL, B.; TORGO, L. Openml: Networked science in machine learning. **SIGKDD Explorations**, ACM, New York, NY, USA, v. 15, n. 2, p. 49–60, 2013. Disponível em: <<http://doi.acm.org/10.1145/2641190.2641198>>. Citado na página 17.

VERONEZE, R. **Tratamento de Dados Faltantes Empregando Biclusterização com Imputação Múltipla**. Dissertação (Mestrado) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 2011. Citado na página 10.

von Luxburg, U.; Schoelkopf, B. Statistical Learning Theory: Models, Concepts, and Results. **ArXiv e-prints**, out. 2008. Citado na página 5.

APÊNDICE A

RESULTADOS PARA OUTRAS BASES DE DADOS

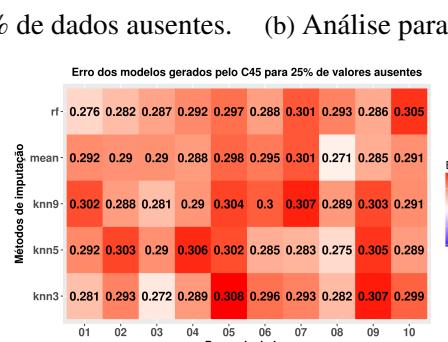
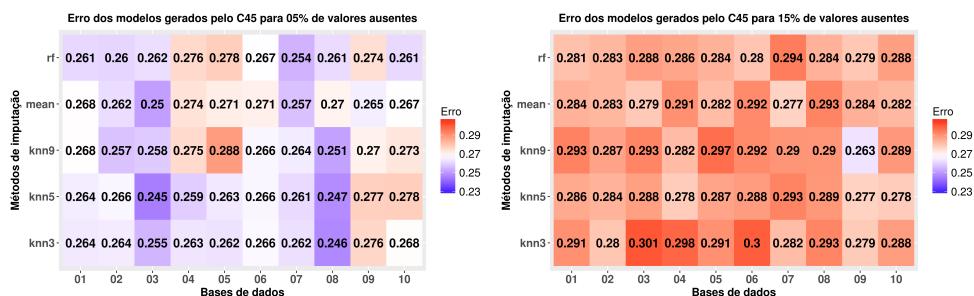
Neste apêndice são mostrados mais resultados obtidos pelo modo de análise. Como já foi feito uma observação geral sobre os dados na Seção 3.5, aqui são descritos os resultados dos modelos gerados para elas.

Na Tabela 10 constam os erros obtidos nas análises para as bases originais. Segue abaixo a lista dos conjuntos de dados aqui descritos, juntamente com os resultados para as bases contendo valores ausentes:

- *Diabetes*: Figuras 15, 16 e 17
- *Ionosphere*: Figuras 18, 19 e 20
- *Iris*: Figuras 21, 22 e 23
- *LED*: Figuras 24, 25 e 26
- *Monk's Problem 1*: Figuras 27, 28 e 29
- *Parkinsons*: Figuras 30, 31 e 32
- *PC Req*: Figuras 33, 34 e 35
- *TAE*: Figuras 36, 37 e 38

Tabela 10 – Informações e análises para bases originais.

| Bases de Dados | Erros para a base original | | |
|--------------------------|----------------------------|---------------|---------------|
| | C4.5 | SVM | RF |
| <i>Diabetes</i> | 0,225 ± 0,011 | 0,238 ± 0,005 | 0,231 ± 0,006 |
| <i>Ionosphere</i> | 0,103 ± 0,013 | 0,059 ± 0,004 | 0,065 ± 0,002 |
| <i>Iris</i> | 0,055 ± 0,008 | 0,037 ± 0,006 | 0,044 ± 0,005 |
| <i>LED</i> | 0,281 ± 0,008 | 0,282 ± 0,006 | 0,287 ± 0,007 |
| <i>Monk's problems 1</i> | 0,020 ± 0,013 | 0,094 ± 0,011 | 0,001 ± 0,001 |
| <i>Parkinsons</i> | 0,152 ± 0,024 | 0,121 ± 0,005 | 0,092 ± 0,006 |
| <i>PC Req</i> | 0,319 ± 0,007 | 0,305 ± 0,004 | 0,301 ± 0,010 |
| <i>TAE</i> | 0,438 ± 0,028 | 0,466 ± 0,022 | 0,367 ± 0,018 |

Figura 15 – Resultados do AM C4.5 para a base *Diabetes*.

(c) Análise para 25% de dados ausentes.

Figura 16 – Resultados do AM RF para a base *Diabetes*.

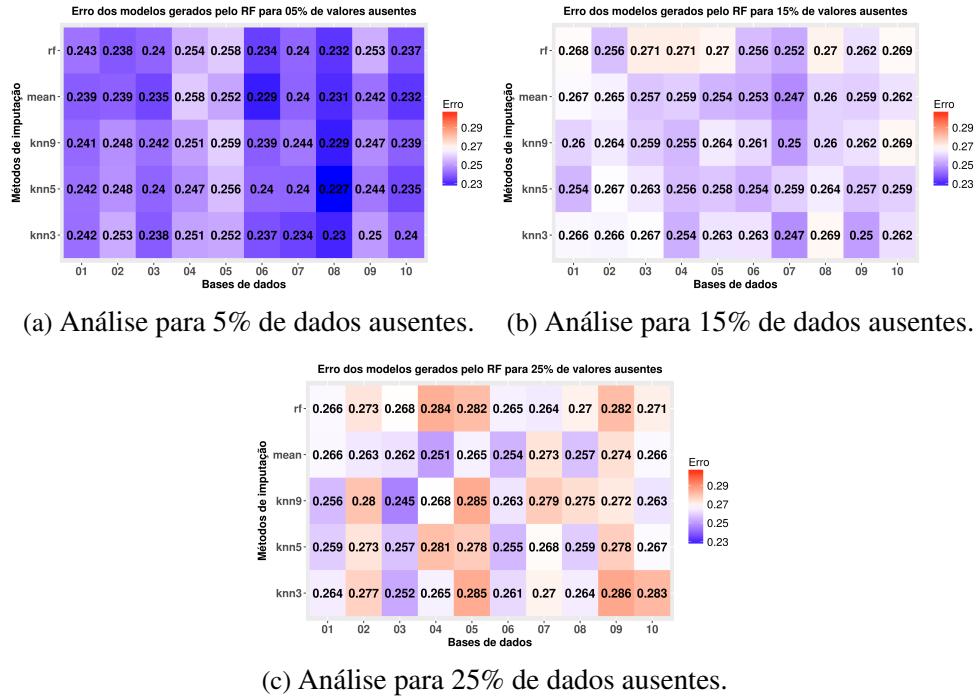


Figura 17 – Resultados do AM SVM para a base *Diabetes*.

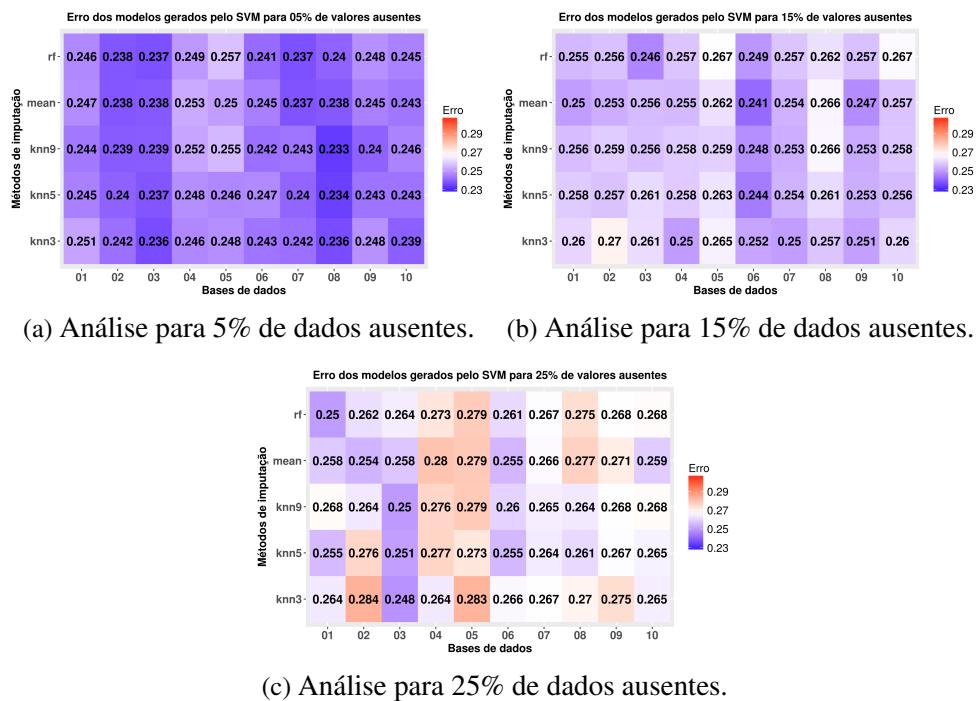
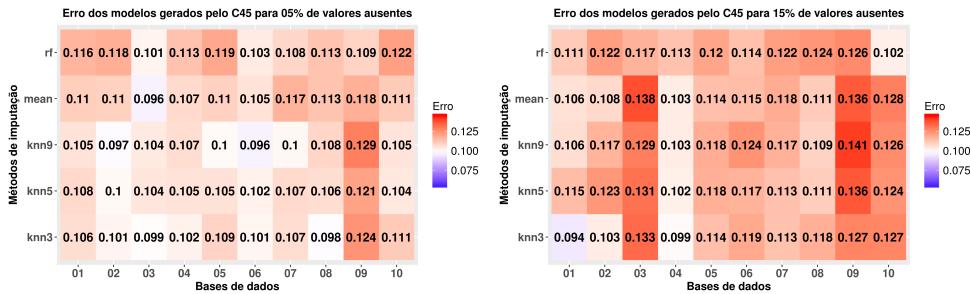
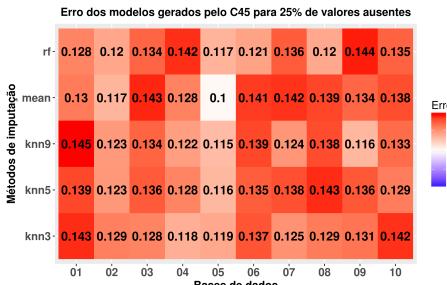
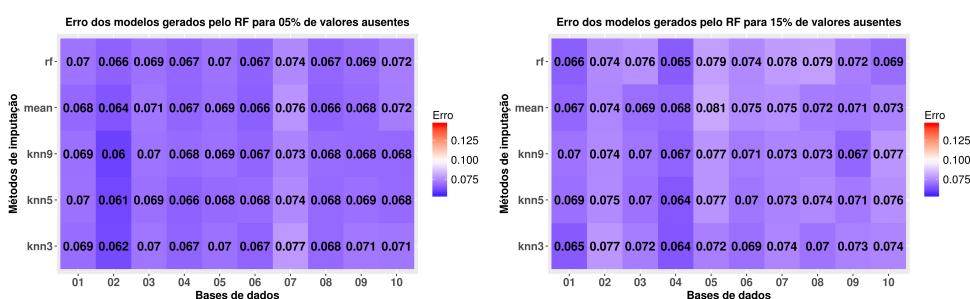
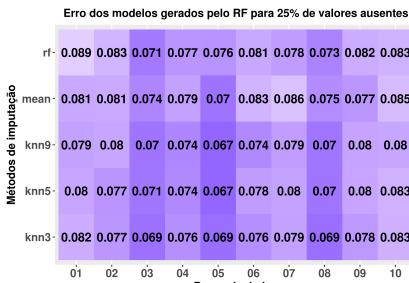


Figura 18 – Resultados do AM C4.5 para a base *Ionosphere*.

(c) Análise para 25% de dados ausentes.

Figura 19 – Resultados do AM RF para a base *Ionosphere*.

(c) Análise para 25% de dados ausentes.



(c) Análise para 25% de dados ausentes.

Figura 20 – Resultados do AM SVM para a base *Ionosphere*.

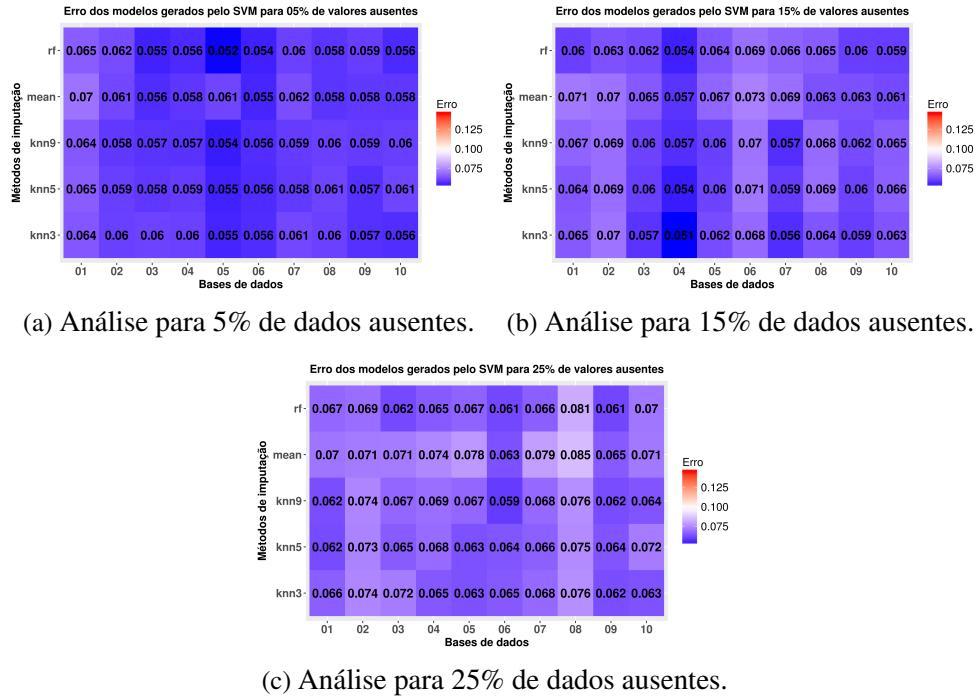


Figura 21 – Resultados do AM C4.5 para a base *Iris*.

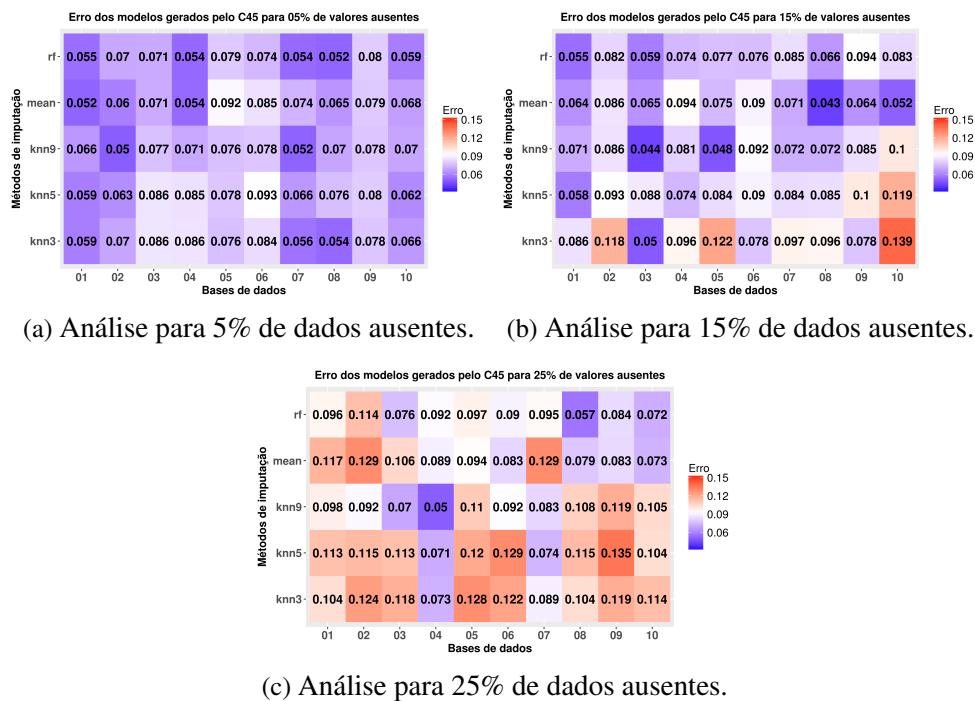
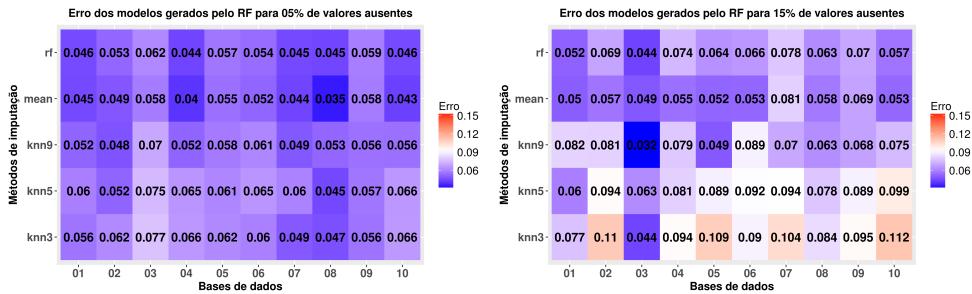
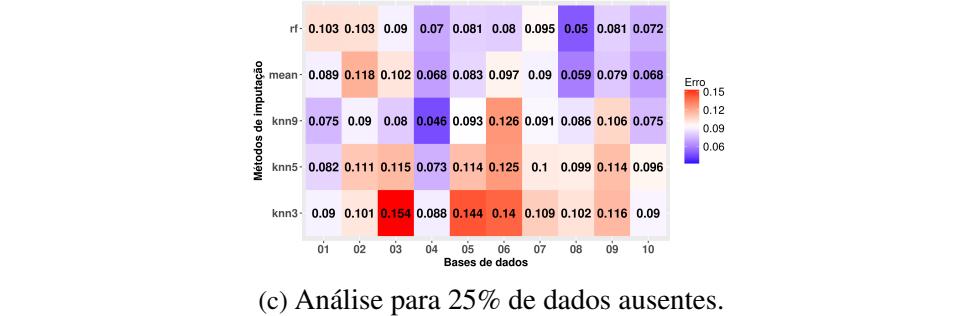


Figura 22 – Resultados do AM RF para a base *Iris*.Figura 23 – Resultados do AM SVM para a base *Iris*.

(a) Análise para 5% de dados ausentes. (b) Análise para 15% de dados ausentes.

(c) Análise para 25% de dados ausentes.

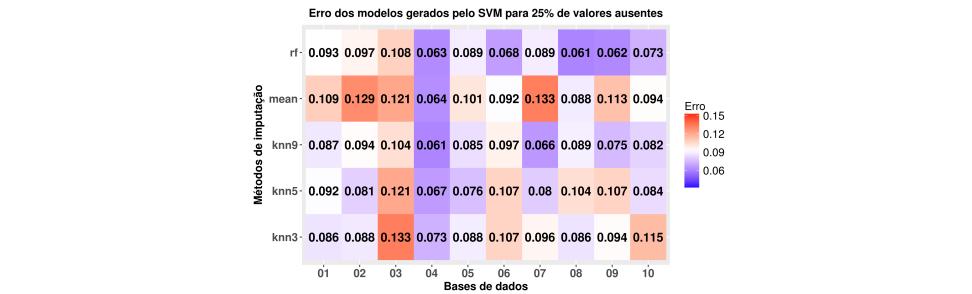


Figura 24 – Resultados do AM C4.5 para a base *LED*.

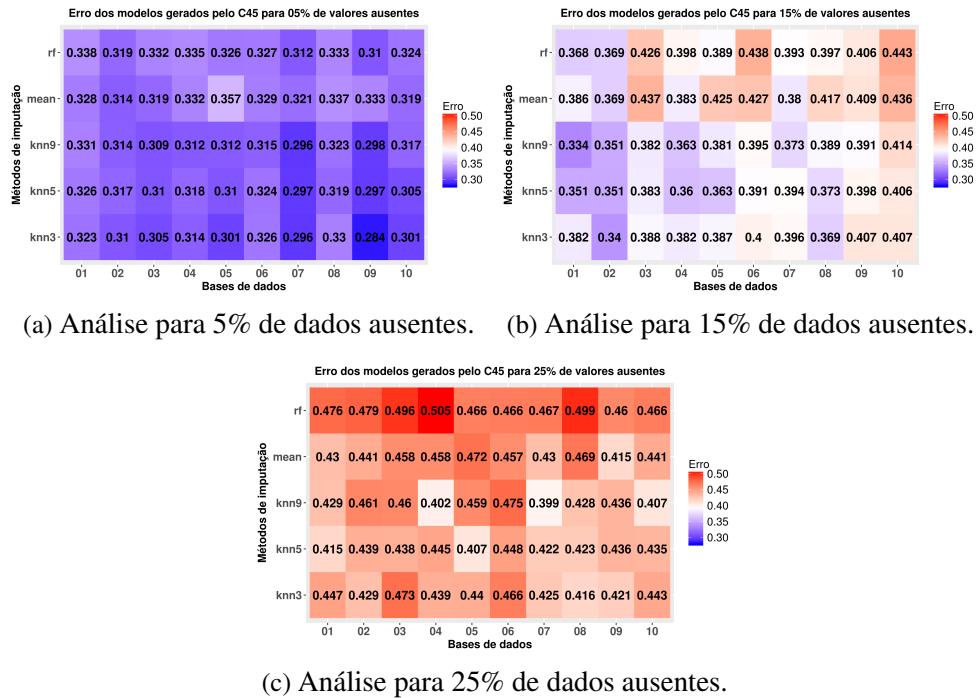


Figura 25 – Resultados do AM RF para a base *LED*.

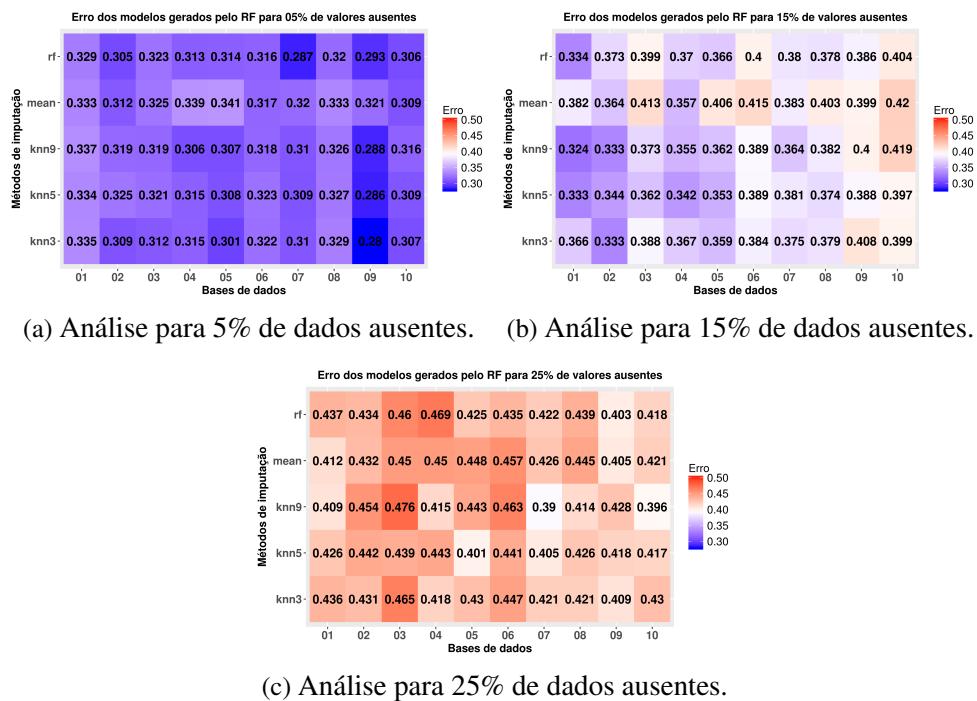


Figura 26 – Resultados do AM SVM para a base LED.

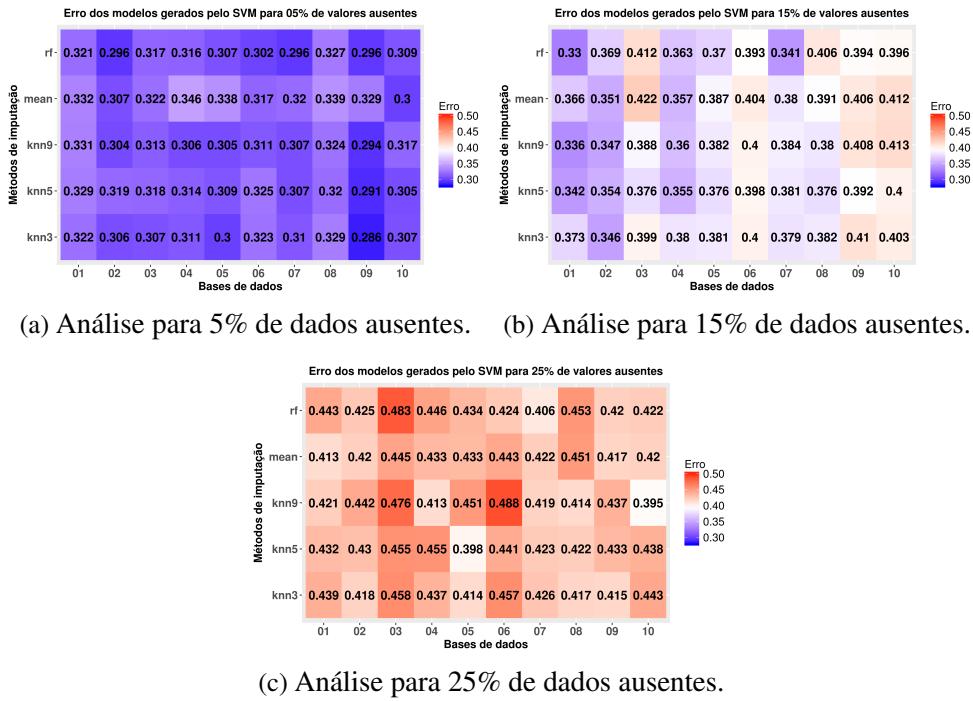


Figura 27 – Resultados do AM C4.5 para a base Monk's Problems 1.

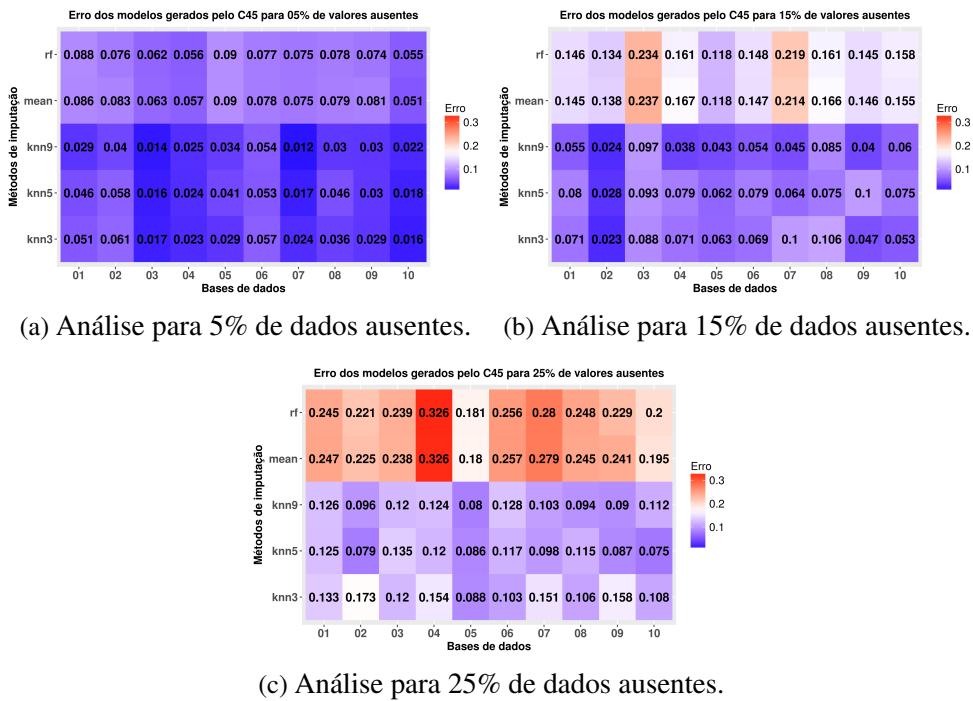


Figura 28 – Resultados do AM RF para a base *Monk's Problems 1*.

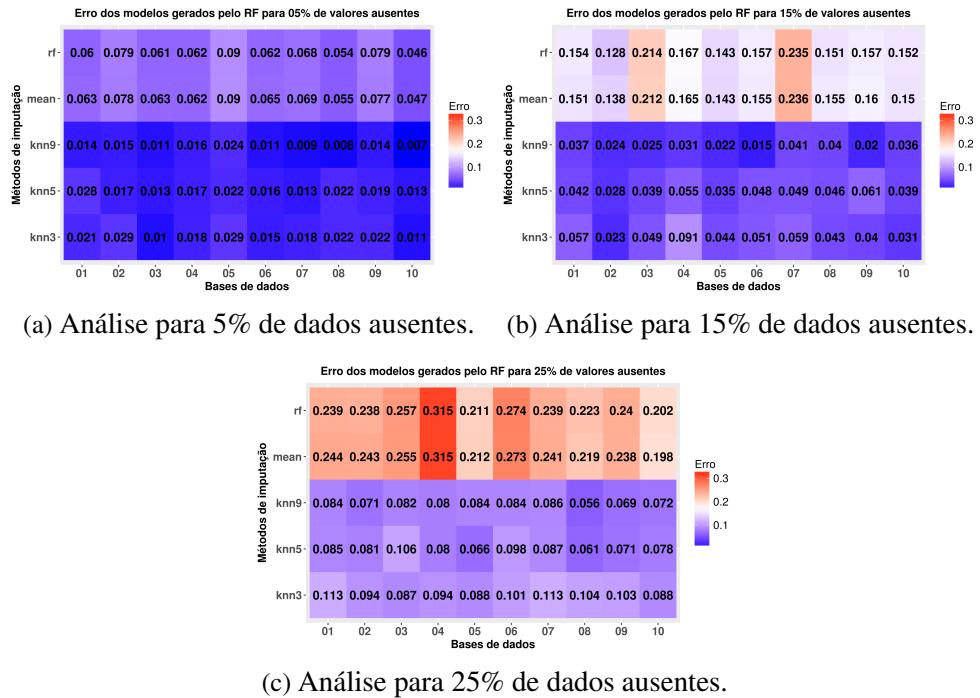


Figura 29 – Resultados do AM SVM para a base *Monk's Problems 1*.

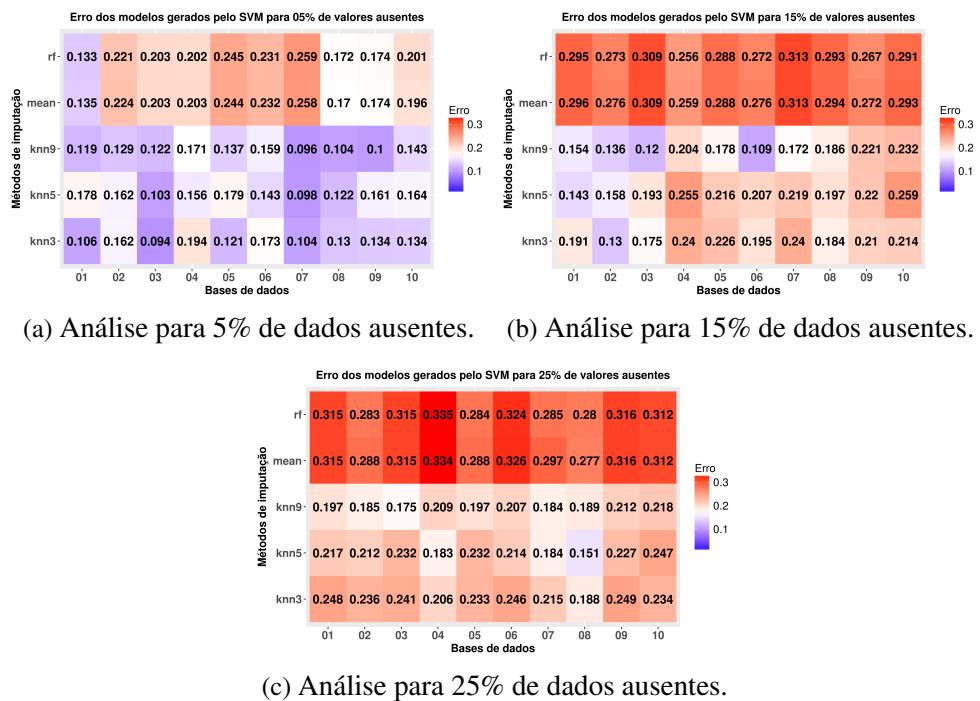
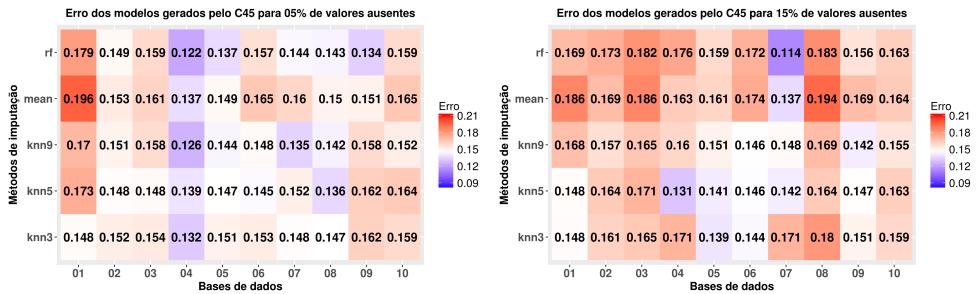
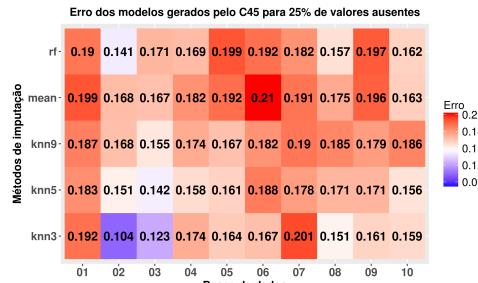
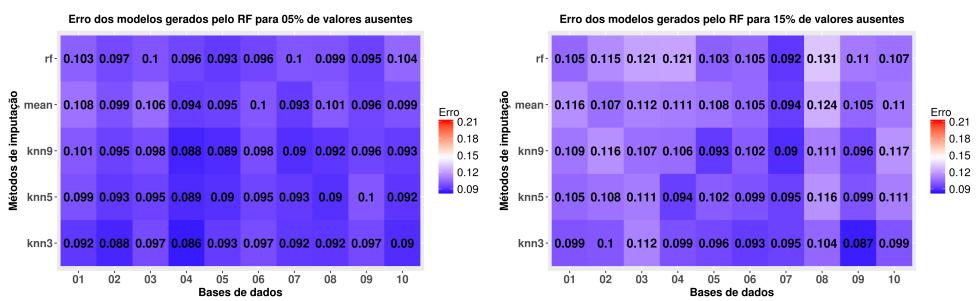


Figura 30 – Resultados do AM C4.5 para a base *Parkinsons*.

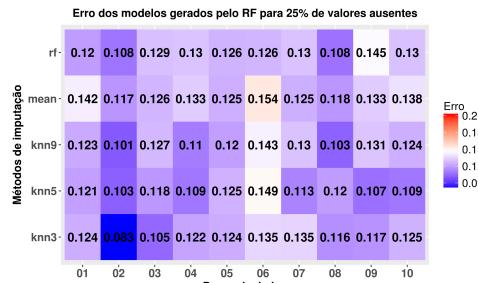
(a) Análise para 5% de dados ausentes. (b) Análise para 15% de dados ausentes.



(c) Análise para 25% de dados ausentes.

Figura 31 – Resultados do AM RF para a base *Parkinsons*.

(a) Análise para 5% de dados ausentes. (b) Análise para 15% de dados ausentes.



(c) Análise para 25% de dados ausentes.

Figura 32 – Resultados do AM SVM para a base *Parkinsons*.

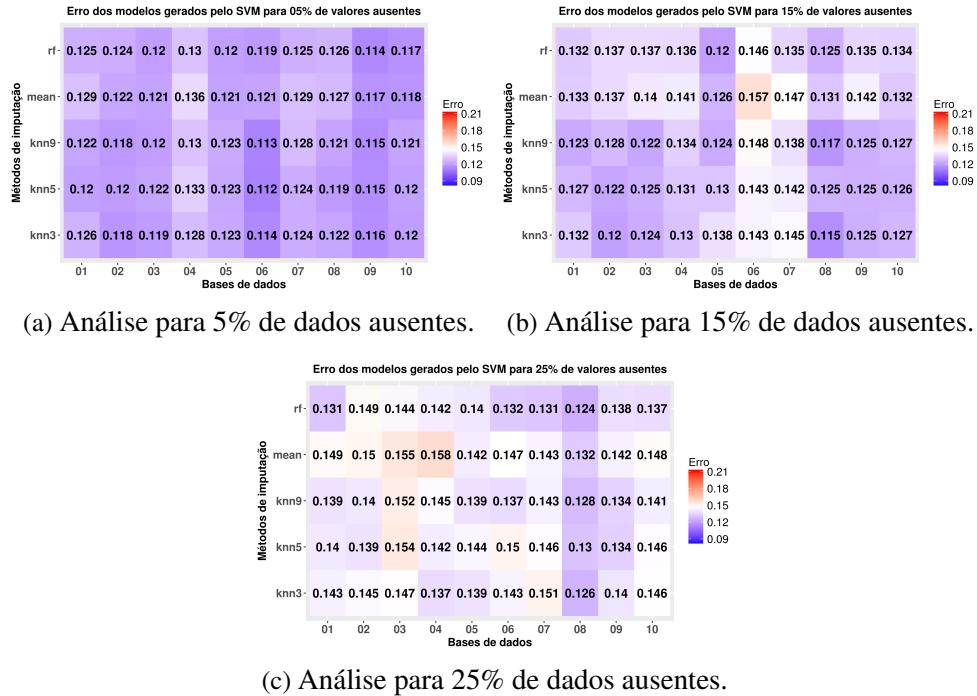


Figura 33 – Resultados do AM C4.5 para a base *PC Req*.

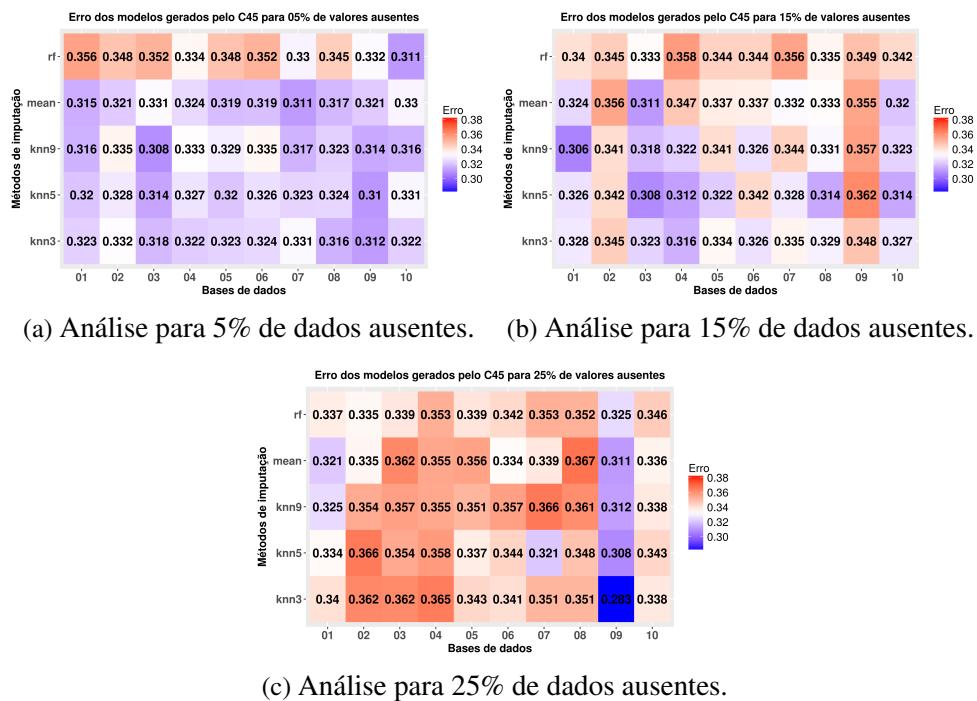


Figura 34 – Resultados do AM RF para a base PC Req.

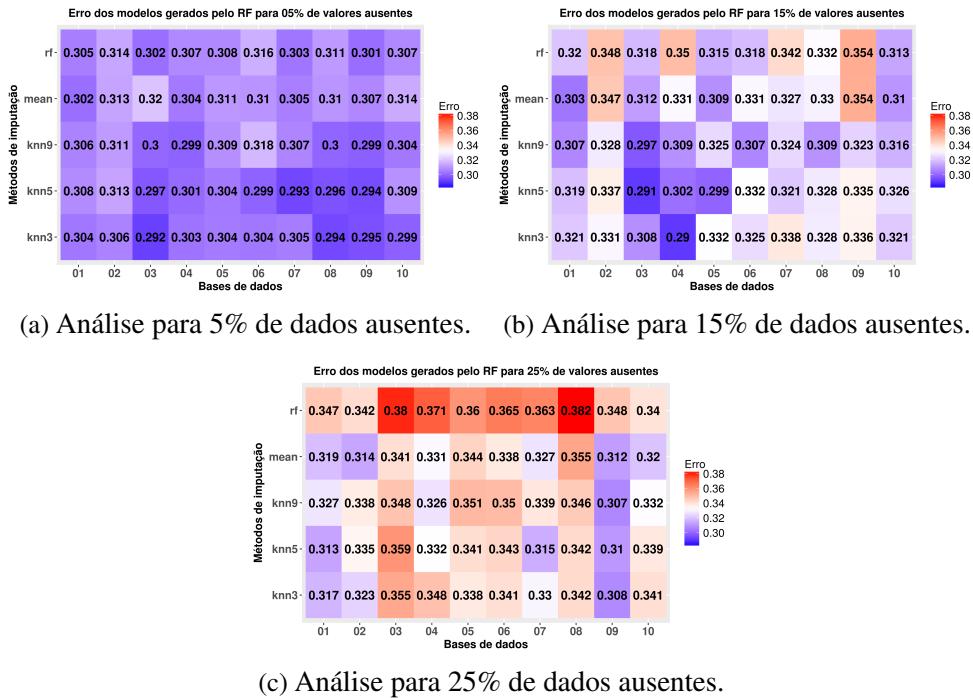


Figura 35 – Resultados do AM SVM para a base PC Req.

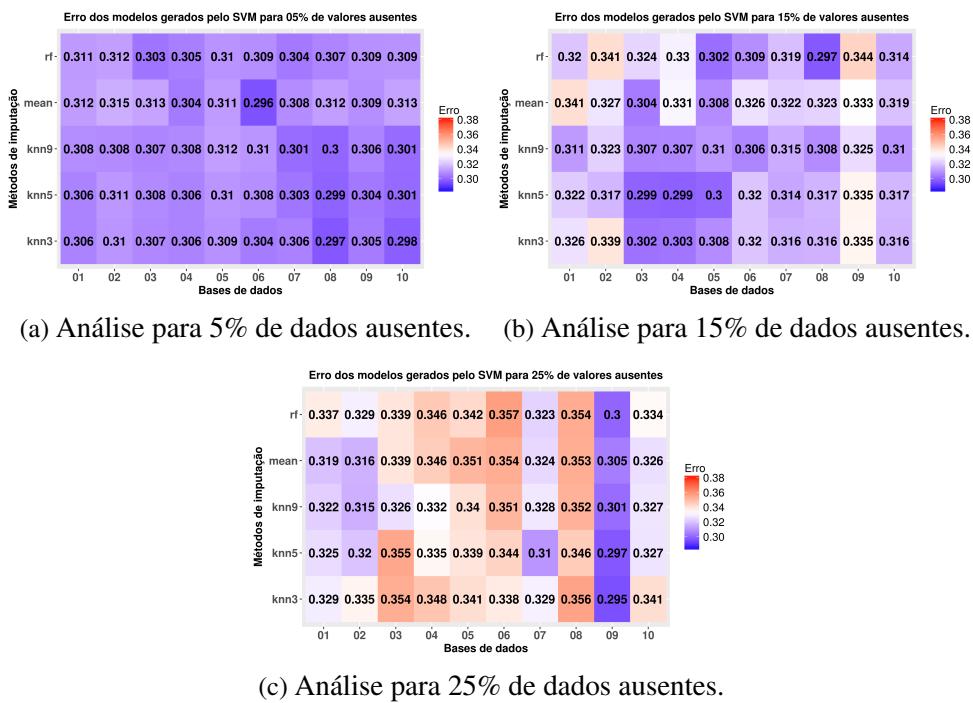


Figura 36 – Resultados do AM C4.5 para a base *TAE*.

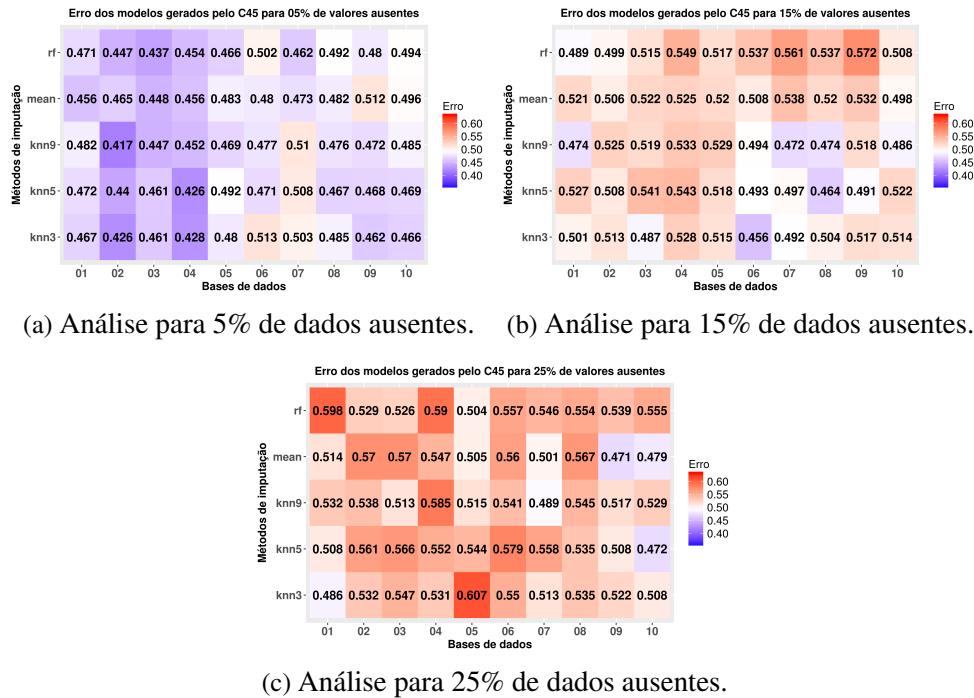


Figura 37 – Resultados do AM RF para a base *TAE*.

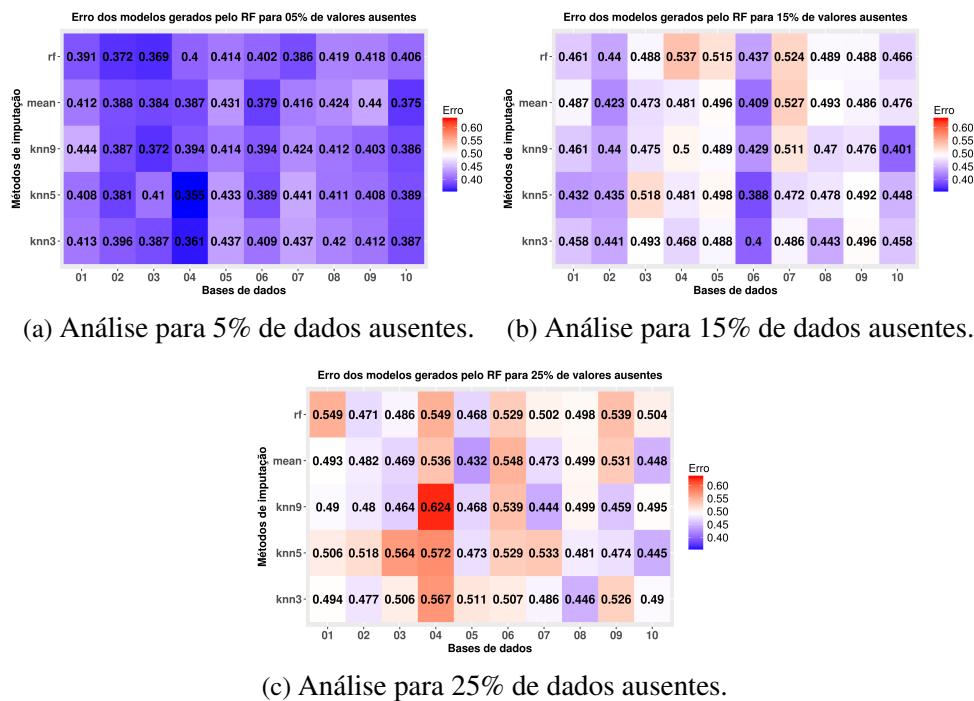


Figura 38 – Resultados do AM SVM para a base TAE.

