UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ADRIANA SCROBOTE

UMA ANÁLISE DA APLICAÇÃO DE ALGORITMOS DE IMPUTAÇÃO DE VALORES FALTANTES EM BASES DE DADOS MULTIRRÓTULO

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2017

ADRIANA SCROBOTE

UMA ANÁLISE DA APLICAÇÃO DE ALGORITMOS DE IMPUTAÇÃO DE VALORES FALTANTES EM BASES DE DADOS MULTIRRÓTULO

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Augusto Foronda

PONTA GROSSA

2017

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa

Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Informática
Bacharelado em Ciência da Computação

TERMO DE APROVAÇÃO

UMA ANÁLISE DA APLICAÇÃO DE ALGORITMOS DE IMPUTAÇÃO DE VALORES FALTANTES EM BASES DE DADOS MULTIRRÓTULO

por

ADRIANA SCROBOTE

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 25 de maio de 2017 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. A candidata foi arguida pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Prof. Dr. Augusto Foronda
Orientador

_____
Prof. MSc. Geraldo Ranthum
Membro titular

_____
Prof(a) Dr(a). Simone de Almeida
Membro titular

_____
Prof. Dr. Ionildo José Sanches
Responsável pelo Trabalho de Conclusão de Curso

_____
Prof. Dr. Erikson Freitas de Morais
Coordenador do curso

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

RESUMO

SCROBOTE, Adriana. **Uma Análise da Aplicação de Algoritmos de Imputação de Valores Faltantes em Bases de Dados Multirrótulo**. 2017. 133 f. Trabalho de Conclusão de Curso Bacharelado em Ciência da Computação - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017.

Há dois tipos de bases de dados que podem ser utilizadas por algoritmos de classificação: monorrótulo e multirrótulo. Em bases de dados monorrótulo cada exemplo está associado a um único rótulo, enquanto que em bases de dados multirrótulo cada exemplo pode estar associado a mais de um rótulo simultaneamente. A ausência de valores em bases de dados multirrótulo é um problema comum e para tentar resolver isso existem os algoritmos de imputação. Algoritmos de imputação de valores faltantes em bases de dados multirrótulo fazem parte da etapa de pré-processamento dos dados para que algoritmos de classificação multirrótulo possam ser aplicados. O tratamento de dados incompletos é feito através da técnica de imputação, onde valores ausentes são substituídos por valores aproximados a partir de outros existentes na base de dados. Existem vários algoritmos que implementam formas de estimar valores. Neste contexto, o presente trabalho faz uma análise da aplicação de algoritmos de imputação de valores omissos em bases de dados multirrótulo para verificar a eficácia de cada um diante de diferentes bases de dados com valores incompletos. Foram testados os algoritmos Imputação pela Moda, Média, Mediana e KNN Iterativo, sendo este último o que obteve os melhores resultados.

Palavras-chave: Base de Dados. Valores Faltantes. Imputação. Classificação Multirrótulo.

ABSTRACT

SCROBOTE, Adriana. **An Analysis of the Application of Missing Values Imputation Algorithms in Multi-label Databases**. 2017. 133 f. Work of Conclusion Course (Graduation in Computer Science) - Federal Technological University of Paraná. Ponta Grossa, 2017.

There are two types of databases that can be used by classification algorithms, which are multi and mono-label databases. In mono-label databases each example is associated with a single label, while in multi-label databases each example may be associated with more than one label simultaneously. The absence of values is a common problem in databases and to solve this problem there are imputation algorithms. Missing values imputation algorithms in multi-label databases are part of the preprocessing data stage so that multi-label classification algorithms can be applied. The treatment of incomplete data is made by imputation, where missing values are substituted by approximate values from other existing values in database. There are several algorithms that implement various ways to estimate values. In this context, the present study is an analysis of the aplication of missing values imputation algorithms in multi-label databases to check the efficiency of each on different databases with incomplete values. The algorithms Imputation by Mode, Mean, Median and Iterative KNN were tested, where the last one got the best results.

Keywords: Database. Missing Values. Imputation. Multi-label Classification.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE GRÁFICOS

SUMÁRIO

1 INTRODUÇÃO

A Inteligência Artificial é uma área importante dentro da computação e é dividida em subáreas, sendo a Aprendizagem de Máquina uma delas (BEZERRA, 2006). Existem dois tipos principais de aprendizagem de máquina: aprendizado não supervisionado e aprendizado supervisionado.

No aprendizado não supervisionado não existe o conhecimento sobre o domínio, já no aprendizado supervisionado há o conhecimento do domínio (BORGES, 2012). O domínio está presente em bases de dados referentes a problemas de aprendizagem de máquina e se trata de um conjunto de respostas esperadas para um determinado problema.

No aprendizado supervisionado há um supervisor que diz se uma previsão do sistema está correta ou não. Assim é possível que durante o processo de aprendizagem o sistema saiba se suas respostas estão coerentes com o resultado esperado e se ajuste a fim de diminuir a taxa de erro para o conjunto selecionado (SILVA, 2008).

Técnicas de aprendizagem de máquina supervisionada são utilizadas para fazer a classificação de dados (BORGES, 2012). O processo de classificação atribui uma classe, também denominada de rótulo para um determinado objeto de acordo com suas características. Esse processo recebe o nome de classificação simples (SILVA, 2014).

Na classificação simples, também denominada de monorrótulo, uma instância do conjunto de dados é associada a somente um rótulo (BORGES, 2012). Outro tipo de classificação é a multirrótulo, onde uma instância pode ser associada a múltiplos rótulos simultaneamente. A classificação multirrótulo possui muitas aplicações, como classificação de textos, categorização de músicas, bioinformática, entre outras (PRATI, 2013).

Um problema comum na utilização de algoritmos de classificação multirrótulo é a ocorrência de valores omissos em base de dados. Uma técnica utilizada para fazer a substituição dos dados faltantes é a imputação de valores que completa a base e permite a análise com todos os dados (NUNES; KLÜCK; FACHEL, 2009).

Existem vários algoritmos que resolvem o problema de valores ausentes em bases de dados, como por exemplo, a Imputação pela Média, Imputação pela

Mediana, entre outros. Por isso, neste trabalho serão feitos experimentos e posteriormente será realizada uma análise da aplicação de algoritmos de imputação de valores em bases de dados multirrótulo a fim de descobrir quais algoritmos apresentam os melhores resultados.

1.1 DESCRIÇÃO DO PROBLEMA

A falta de dados é um problema frequente na investigação científica, fator que dificulta a extração de informações importantes presentes em grandes bases de dados. O problema de dados omissos pode se dar por diversos motivos, como: falhas de digitação de itens importantes, erros no preenchimento de tabelas, problemas na digitação de documentos, perda de dados ao longo dos anos, entre outros (NUNES; KLÜCK; FACHEL, 2009).

É importante solucionar o problema de bases de dados com valores incompletos, pois para algoritmos de classificação multirrótulo que utilizam aprendizagem de máquina supervisionada o tratamento é indispensável, já que muitos dos algoritmos existentes na literatura podem ser aplicados somente em bases de dados completas.

O tratamento de valores ausentes deve ser planejado, caso contrário, podem ser introduzidas distorções no conhecimento afetando o desempenho do algoritmo de classificação multirrótulo. O ideal é que antes de ocorrer o processo de classificação, ocorra o pré-processamento dos dados para que problemas com bases de dados, como valores desconhecidos e incompletos sejam resolvidos (BATISTA, 2003).

Determinar valores para um conjunto de dados omissos não é uma tarefa simples, já que o uso de métodos inadequados pode levar a resultados incorretos. Desde a década de 80, estão sendo desenvolvidas técnicas para substituir dados faltantes por estimativas de valores admissíveis para serem inseridos (NUNES, 2007). Diante de vários algoritmos que resolvem o problema de valores ausentes em bases de dados multirrótulo, torna-se interessante saber quais algoritmos se destacam pelos resultados alcançados. Esta verificação pode ser feita através de uma análise da aplicação destes algoritmos.

1.2 OBJETIVOS

Este trabalho tem por objetivo geral desenvolver uma análise da aplicação de algoritmos de imputação de valores faltantes em bases de dados multirrótulo para a utilização da tarefa de classificação.

1.2.1 Objetivos Específicos

Este trabalho tem como objetivos específicos:

- Compreender a classificação multirrótulo;
- Identificar características de bases de dados multirrótulo com valores faltantes;
- Compreender o processo de imputação de valores;
- Realizar testes com os algoritmos de imputação de valores em bases de dados multirrótulos;
- Analisar matematicamente os resultados obtidos.

1.3 JUSTIFICATIVA

A classificação multirrótulo está sendo utilizada em diversas áreas, como na categorização de textos, predição de proteínas, classificação de músicas, entre outras.

A categorização de textos é o processo de atribuir automaticamente uma ou mais categorias para um determinado texto. Com o surgimento de bibliotecas digitais e o aumento significativo destes repositórios é importante melhorar o processo de recuperação de informações, pois a classificação manual torna-se inviável. (BORGES, 2012).

Os documentos são textos recuperados através de uma pesquisa feita pelo usuário a partir de palavras-chave. É comum em uma pesquisa serem recuperados documentos irrelevantes para o usuário. Com algoritmos de classificação multirrótulo pode-se melhorar o processo de categorização de textos atingindo o objetivo principal que é recuperar o maior número de documentos relevantes para o usuário.

Entretanto, diversas bases de dados utilizadas em algoritmos de classificação multirrótulo apresentam valores faltantes de seus atributos que podem diminuir o desempenho do algoritmo e causar a geração de resultados errôneos, dificultando possíveis análises. Uma maneira de resolver este problema é através da imputação, ou seja, da atribuição de valores estimados a partir de outros existentes na base de dados (OLIVEIRA, 2009).

De modo geral, algoritmos de classificação multirrótulo dependem do tratamento correto dos dados para poder atingir todo o seu potencial, ou seja, dependem de algoritmos de imputação de valores faltantes em base de dados multirrótulo. Neste contexto, surge a necessidade de analisar algoritmos existentes de imputação aplicados em bases de dados multirrótulo, verificando quais deles são eficientes e que podem melhorar os resultados obtidos por algoritmos de classificação multirrótulo, problema que será tratado neste trabalho.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho divide-se em cinco capítulos. O capítulo 2 possui explicações mais aprofundadas sobre conceitos básicos de classificação, classificação multirrótulo, medidas de avaliação de algoritmos de classificação multirrótulo, imputação, técnicas de imputação e também apresenta trabalhos relacionados. O capítulo 3 explica os experimentos realizados neste trabalho bem como a metodologia utilizada. O capítulo 4 apresenta os resultados obtidos com os experimentos realizados. Por fim, o capítulo 5 traz as considerações finais.

2 REFERENCIAL TEÓRICO

Neste Capítulo são apresentados conceitos básicos das tarefas de classificação e imputação. Na Seção 2.1 são explicados os principais conceitos, métodos e medidas de avaliação referentes à classificação monorrótulo. Na Seção 2.2 são descritos os principais conceitos, métodos e medidas de avaliação de classificadores multirrótulo. A Seção 2.3 traz as características dos dados com os principais problemas existentes nos valores utilizados por algoritmos de classificação. Na Seção 2.4 é apresentado o processo de imputação com seus principais métodos. Por fim, a Seção 2.5 expõe os trabalhos relacionados com o tema do presente trabalho.

2.1 CONCEITOS BÁSICOS DE CLASSIFICAÇÃO

A aprendizagem de máquina é uma área da Inteligência Artificial que busca desenvolver técnicas e algoritmos que permitam ao computador aprender (AMORIN; BARONE; MANSUR, 2008).

Utiliza-se aprendizagem de máquina para resolver problemas que são difíceis de obter uma solução manualmente, pois envolvem uma grande quantidade de dados para serem analisados. A classificação é uma das principais tarefas que utilizam aprendizagem de máquina (CERRI, 2010).

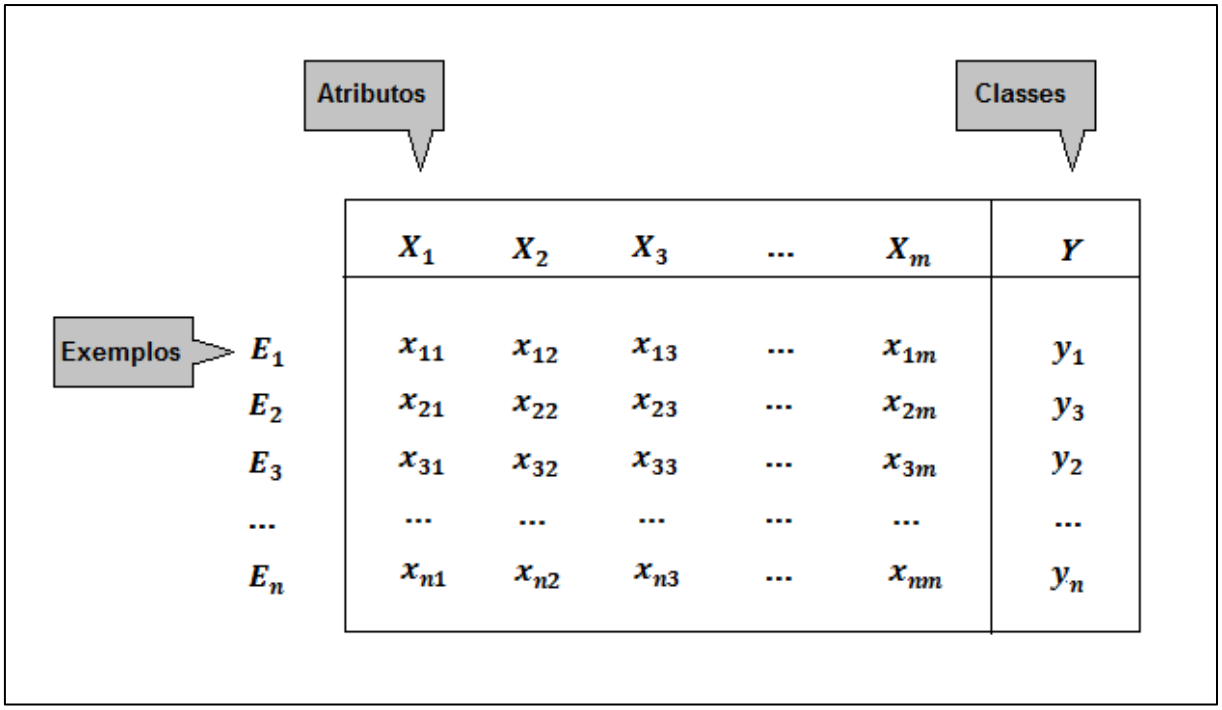
O processo de classificação de dados associa um exemplo a uma ou mais classes de acordo com suas características. Em aprendizagem de máquina, a classificação faz parte do tipo de aprendizagem supervisionada, a qual constrói algoritmos de indução com o objetivo de conseguir um bom classificador a partir de um conjunto de exemplos previamente rotulados. O classificador resultante pode, então, ser utilizado para fazer a classificação de novos exemplos que ainda não possuem rótulos (BORGES, 2012). Neste processo alguns conceitos são importantes:

- Atributo: valor utilizado para descrever uma característica de um exemplo;

- Classe: também conhecida como rótulo, descreve o atributo-alvo, ou seja, o resultado esperado ao realizar um processo de classificação para um determinado exemplo;
- Exemplo: é um conjunto fixo de atributos do qual um classificador será construído ou utilizado, é também conhecido como instância;
- Conjunto de exemplos: é composto por vários exemplos com seus respectivos atributos. Também é conhecido como conjunto de dados ou conjunto de instâncias.

Na Figura 1 pode-se ver a representação de um conjunto de dados, onde são exemplificados os conceitos de atributo, classe e exemplo.

Figura 1 - Representação de um conjunto de dados



Fonte: Autoria própria

Na Figura 1, exemplos são representados por linhas, ou seja, há *n* exemplos neste conjunto. Cada atributo é representado por uma coluna. Portanto, existem *m* atributos que descrevem as características de cada exemplo. Por fim, existe uma coluna especial que representa a classe que cada exemplo está relacionado.

Para que a classificação possa ser feita, inicialmente os dados são preparados em um conjunto de exemplos. Cada exemplo é representado por uma tupla de atributos, sendo um destes chamado de rótulo da classe, ou simplesmente

classe. Os valores dos atributos descrevem características da instância e o atributo de rótulo da classe representa a saída esperada para o exemplo, pois cada instância pertence a uma classe pré-estabelecida (CERRI, 2010).

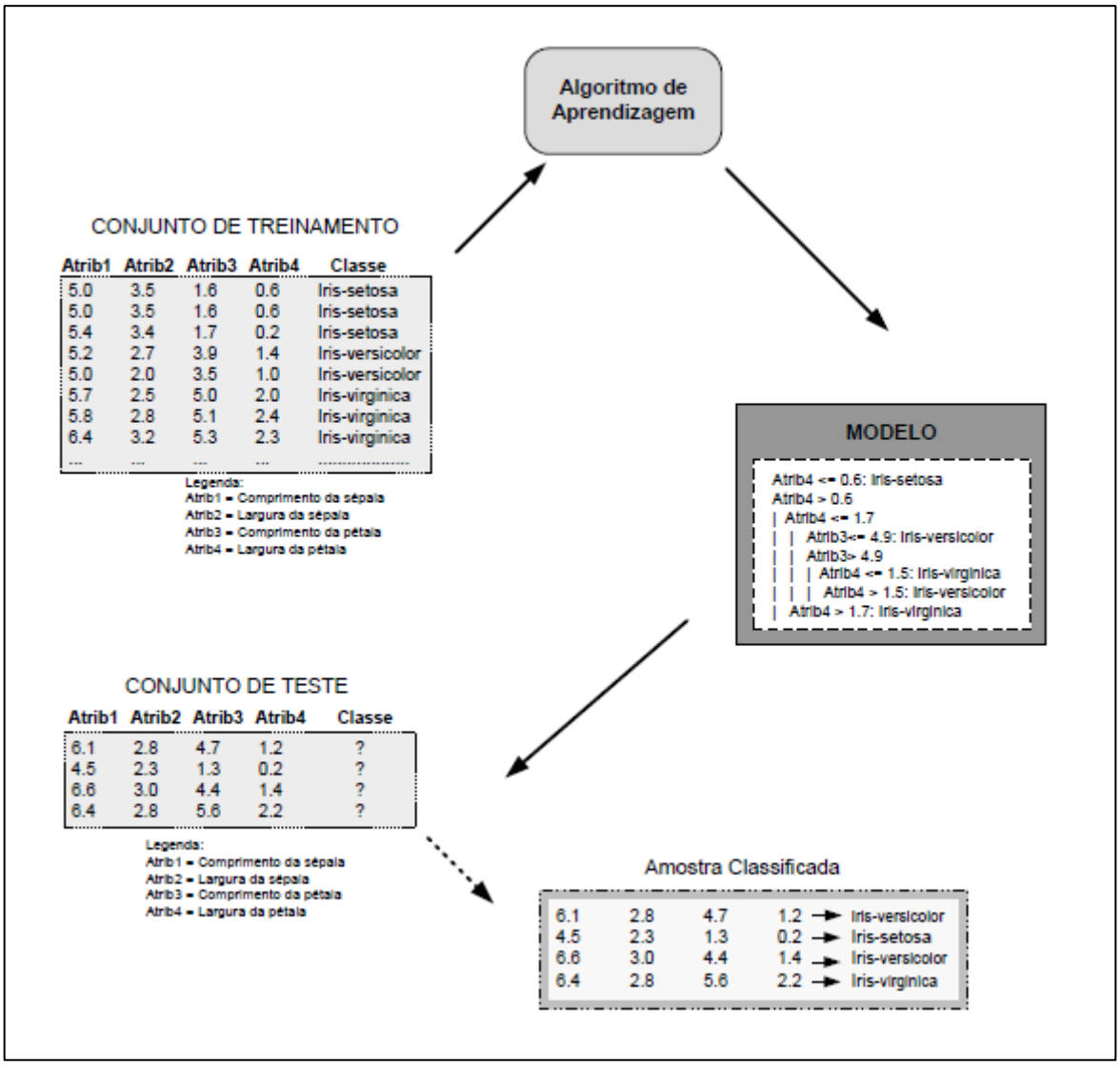
Os dados da tarefa de classificação são processados em duas fases: treinamento e teste. Após o preparo dos dados, o conjunto de exemplos com rótulos pré-estabelecidos (conjunto de treinamento) é cedido para um algoritmo de treinamento (indutor) para que a aprendizagem do classificador seja feita. A construção do classificador é feita a partir dos valores das instâncias fornecidas para a fase de treinamento, por isso o conjunto de exemplos fornecido deve representar bem o domínio da aplicação (REZENDE, 2005).

Depois da execução da fase de treinamento, obtém-se um modelo resultante que é um classificador capaz de associar uma classe para exemplos desconhecidos que não possuem rótulos (CERRI, 2010).

Para verificar a eficiência do modelo obtido pelo algoritmo de treinamento, tem-se a fase de teste também chamada de fase de validação, na qual é necessário disponibilizar um segundo conjunto de dados que não tenham participado na construção do classificador para que ocorra o processo de classificação e, então é possível medir o desempenho do classificador, ou seja, verificar sua taxa de acertos para os rótulos atribuídos para as novas instâncias em comparação com seus rótulos esperados (AMORIM; BARONE; MANSUR, 2008).

Na Figura 2 é possível ver o funcionamento completo do processo de classificação de dados. Inicialmente, um conjunto de exemplos rotulados (ou conjunto de treinamento) é fornecido ao algoritmo de treinamento para que seja feita a construção de um classificador utilizando aprendizagem de máquina. Após terminar a fase de treinamento obtém-se um classificador, também chamado de modelo que servirá para fazer a classificação de novas instâncias que não foram utilizadas na fase de treinamento. Por fim, pode-se fazer a validação do classificador através da fase de teste, onde é fornecido ao algoritmo de teste um novo conjunto de dados com rótulos desconhecidos para que todos os exemplos sejam classificados, ou seja, recebam um rótulo.

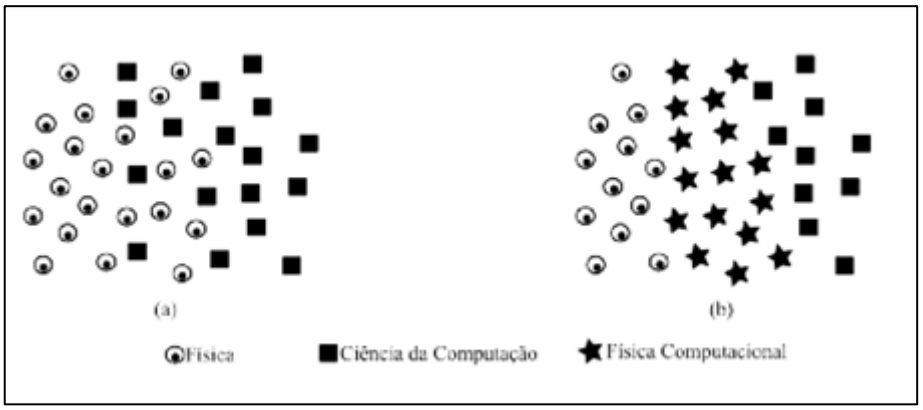
Figura 2 - Classificação de dados



Fonte: Borges (2012)

Considerando a quantidade de rótulos associados para cada um dos exemplos do conjunto de dados fornecidos ao algoritmo de treinamento, existem dois tipos de problemas de classificação: monorrótulo ou multirrótulo. Se os exemplos do conjunto de treinamento estão associados a somente um rótulo, então se trata de classificação tradicional, chamada de monorrótulo ou unirrótulo. Por outro lado, se os exemplos estão associados a mais de um rótulo simultaneamente, o problema de classificação é multirrótulo (RODRIGUES, 2014). A diferença entre classificação monorrótulo e multirrótulo pode ser visualizada na Figura 3.

Figura 3 - Problema de classificação monorrótulo e multirrótulo



Fonte: Cerri (2010)

A Figura 3 ilustra exemplos que fazem parte de classificação monorrótulo (a) e multirrótulo (b). Na classificação monorrótulo, as instâncias estão associadas a somente um rótulo, ou seja, neste caso pertencem a classe “Física” ou “Ciência da Computação”. Por outro lado, na classificação multirrótulo as instâncias que pertencem às classes “Física” e “Ciência da Computação” simultaneamente são classificadas com a classe “Física Computacional”.

2.1.1 Métodos de Classificação Monorrótulo

Na literatura há vários métodos de classificação monorrótulo que visam atribuir uma classe para exemplos não rotulados. No entanto, neste capítulo serão apresentados somente três métodos: KNN, *Naive Bayes* e J48. Uma vez que o foco deste trabalho não é classificação monorrótulo, mas sim imputação que abrange a classificação multirrótulo. O KNN foi escolhido porque é um algoritmo muito utilizado para resolver problemas que vão além da classificação monorrótulo. O KNN pode sofrer adaptações quando são necessárias para participar dos processos de imputação e classificação multirrótulo. Os algoritmos *Naive Bayes* e J48 servirão como classificadores bases para os classificadores multirrótulo apresentados na Seção 2.2.1.

2.1.1.1 KNN

O *k-Nearest-Neighbor* (KNN) também é conhecido como o algoritmo dos *k* vizinhos mais próximos. O KNN recebe um conjunto de dados que é utilizado durante a etapa de treinamento e também na etapa de teste para classificar um novo exemplo (RODRIGUES, 2014).

Para atribuir um rótulo a um novo exemplo, deve ser calculada a similaridade deste exemplo em relação a todos os outros existentes no conjunto de dados, através de uma medida de distância. Geralmente, é utilizada a Distância Euclidiana (1) que resulta na distância entre os pontos $P = (p_1, p_2, \ldots, p_n)$ e $Q = (q_1, q_2, \ldots, q_n)$.

$$d_{(p,\ q)} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{1}$$

Na equação (1):

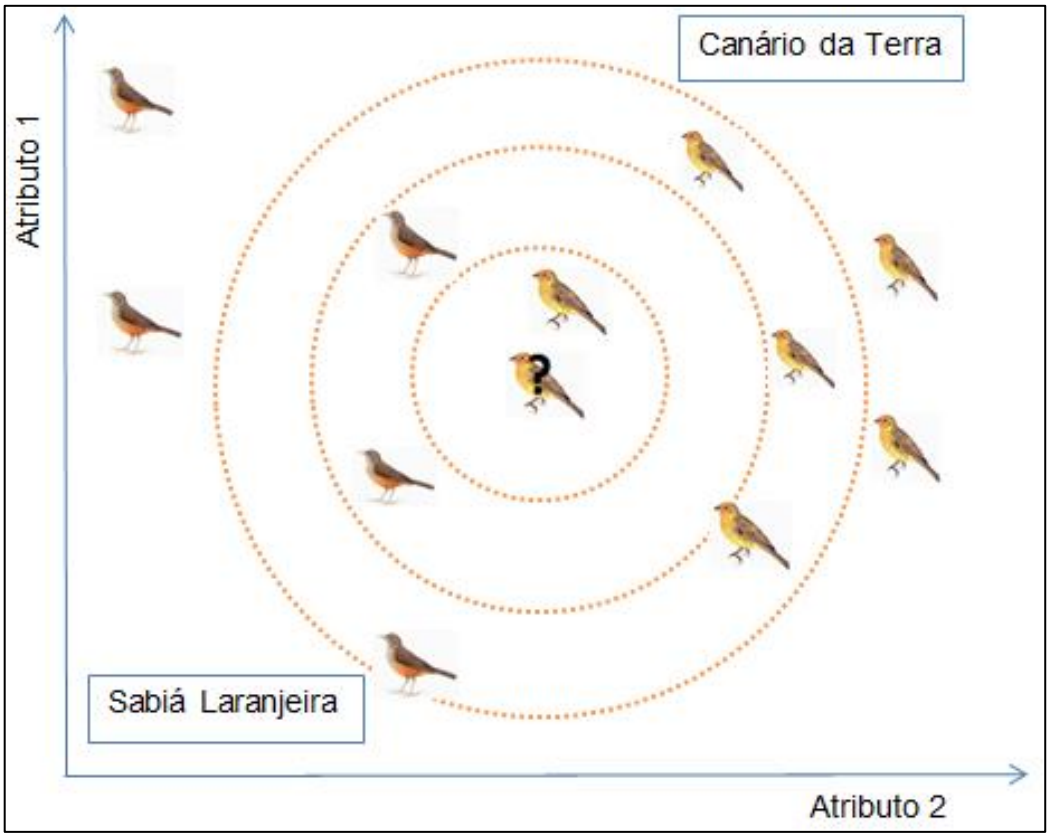
- $d_{(p,\ q)}$ é a Distância Euclidiana entre os pontos P e Q num espaço n-dimensional;
- n é o número de dimensões;
- p_i é o valor do ponto P na dimensão i;
- q_i é o valor do ponto Q na dimensão i.

Após serem calculadas as Distâncias Euclidianas para um exemplo desconhecido, são selecionados os *k* vizinhos mais próximos, ou seja, os que possuem menor distância em relação ao novo exemplo, sendo *k* um parâmetro fornecido ao algoritmo que representa a quantidade de vizinhos mais próximos a serem considerados (CERRI, 2010). O rótulo que mais se repete entre os *k* vizinhos mais próximos selecionados é o rótulo atribuído ao novo exemplo.

Na Figura 4 é mostrado o impacto do valor de *k* no resultado do algoritmo KNN aplicado em um problema de classificação monorrótulo para pássaros, no qual um novo exemplo pode ser classificado com apenas uma espécie de pássaro, sendo “Sabiá Laranjeira” ou “Canário da Terra”. Para classificar um pássaro com espécie desconhecida, são calculadas as distâncias de todos os pássaros existentes em relação ao novo pássaro através da Distância Euclidiana. Em seguida, são

encontrados os *k* vizinhos mais próximos e então é realizada a classificação com o rótulo que mais apareceu.

Figura 4 - Impacto do valor de *k* no algoritmo KNN



Fonte: Adaptado de Cerri (2010)

Na Figura 4, aplicando o algoritmo KNN com *k* = 1, o pássaro desconhecido será classificado como “Canário da Terra”. Para *k* = 3, o pássaro será classificado como pertencente à classe “Sabiá Laranjeira”, enquanto que para *k* = 7, a classe atribuída será “Canário da Terra”.

2.1.1.2 *Naive Bayes*

O *Naive Bayes* é um algoritmo de classificação probabilístico que resulta em um modelo que contém um conjunto de probabilidades estimado através da contagem da frequência dos valores fornecidos em uma base de dados (PATIL; SHEREKAR, 2013).

Para entender o funcionamento do classificador *Naive Bayes* é necessário o conhecimento prévio de Probabilidade Condicional e Teorema de *Bayes*, pois o algoritmo utiliza esses dois conceitos.

Probabilidade Condicional é a probabilidade de ocorrer determinado evento sob uma dada condição. Em outras palavras, probabilidade condicional é a probabilidade de ocorrer o evento *B* sob a condição de *A* ter ocorrido. Em Probabilidade Condicional, eventos são dependentes, ou seja, a ocorrência do primeiro evento afeta o resultado do segundo evento. (VIEIRA, 2011).

Para exemplificar o conceito de Probabilidade Condicional, considere a seguinte pergunta: qual é a probabilidade de um dia estar ventando forte e com umidade alta dado que houve jogo de tênis? Nesse caso, o conhecimento prévio do resultado do primeiro evento que é o acontecimento do jogo de tênis, influencia o segundo resultado. Portanto, Probabilidade Condicional é a probabilidade de ocorrer determinado evento, mas já sabendo qual evento ocorreu antes.

Utilizando a Probabilidade Condicional é possível chegar a uma evidência a partir de um resultado conhecido, mas muitas vezes é preciso calcular o inverso para encontrar a chance de um resultado acontecer, dada uma evidência. Para exemplificar considere a seguinte questão: qual é a probabilidade de ter ocorrido jogo de tênis dado que durante o dia ventou forte e a umidade estava alta? Nesses casos em que o evento *A* antecede o evento *B* e onde deve-se considerar que o evento *B* já tenha ocorrido utiliza-se o Teorema de *Bayes*, apresentado na equação (2).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad (2)$$

Na equação (2):

- P(*A*|*B*): probabilidade de ocorrer o evento *A*, dado que o evento *B* já tenha ocorrido;
- P(*B*|*A*): probabilidade de ocorrer o evento *B*, dado que o evento *A* já tenha ocorrido;
- P(*A*): probabilidade de ocorrer o evento *A*;
- P(*B*): probabilidade de ocorrer o evento *B*.

O Quadro 1 representa um problema monorrótulo com 14 exemplos de dias utilizado para saber se houve jogo de tênis ou não dependendo dos valores dos atributos Vento e Umidade. Este problema será usado para exemplificar a execução do algoritmo Naive Bayes.

Quadro 1 - Base Jogo de Tênis

Umidade	Vento	Jogou Tênis
Alta	Fraco	Não
Alta	Forte	Não
Alta	Fraco	Sim
Alta	Fraco	Sim
Normal	Fraco	Sim
Normal	Forte	Não
Normal	Forte	Sim
Alta	Fraco	Não
Normal	Fraco	Sim
Normal	Fraco	Sim
Normal	Forte	Sim
Alta	Forte	Sim
Normal	Fraco	Sim
Alta	Forte	Não

Fonte: Adaptado de Rodrigues (2014)

O Quadro 2 é um Quadro de Frequência referente ao problema de Jogo de Tênis apresentado no Quadro 1. O Quadro de Frequência é construído a partir do conjunto de treinamento fornecido ao algoritmo Naive Bayes, neste caso a Base Jogo de Tênis.

Quadro 2 - Quadro de Frequência base Jogo de Tênis

Jogou Tênis?	Vento		Umidade		Quantidade Exemplos
	Forte	Fraco	Alta	Normal	
Sim	3	6	3	6	9
Não	3	2	4	1	5
Total	6	8	7	7	14

Fonte: Autoria própria

O Quadro de Frequência apresentado no Quadro 2 mostra que existem ao todo 14 exemplos no problema de Jogo de Tênis, sendo 9 classificados como Sim e

5 como Não. Também são demonstradas as quantidades de exemplos que possuem Vento Forte, Vento Fraco, Umidade Alta e Umidade Normal quando a classe atribuída for Sim e também quando for Não. Por fim, são apresentados os totais de exemplos com Vento Forte, Vento Fraco, Umidade Alta e Umidade Normal.

Para classificar, ou seja, responder se houve ou não jogo de tênis para um exemplo desconhecido dado que ele possui Vento Forte e Umidade Alta deve ser utilizado o Quadro de Frequência para calcular as probabilidades para este exemplo ser classificado como Sim ou Não.

O cálculo de probabilidade clássica para um evento *A* acontecer se dá pela equação (3), onde $n(\Omega)$ representa o número total de resultados igualmente possíveis e $n(A)$ é o número de resultados favoráveis ao evento *A* (VIEIRA, 2011).

$$P(A) = \frac{n(A)}{n(\Omega)} \tag{3}$$

Aplicando a equação (3) para encontrar as probabilidades de um exemplo ser classificado como Sim ou Não, tem-se:

P(Sim) = 9 / 14 = 0,64
P(Não) = 5 / 14 = 0,36

Também é necessário calcular as probabilidades de um exemplo possuir Vento Forte e Umidade Alta, através da equação (3):

P(Forte) = 6 / 14 = 0,43
P(Alta) = 7 / 14 = 0,5

Agora são calculadas as probabilidades condicionais, utilizando a equação (3), para um dia ter vento forte e umidade alta, dado se houve ou não jogo de tênis:

P(Forte | Sim) = 3 / 9 = 0,33
P(Alta | Sim) = 3 / 9 = 0,33
P(Forte | Não) = 3 / 5 = 0,6
P(Alta | Não) = 4 / 5 = 0,8

Após todas as probabilidades serem calculadas, pode-se aplicar o Teorema de *Bayes*. Acreditando na hipótese de que o exemplo desconhecido será classificado com Sim, então o evento *A* da equação (2) será substituído por Sim e o evento *B* por Forte e Alta, portanto:

$$P(Sim \mid Forte, Alta) = [P(Forte \mid Sim) * P(Alta \mid Sim) * P(Sim)] / P(Forte) * P(Alta)$$
$$P(Sim \mid Forte, Alta) = (0,33 * 0,33 * 0,64) / (0,43 * 0,5) = 0,32$$

Agora, acreditando na hipótese de que o exemplo desconhecido será classificado com Não, então o evento *A* da equação (2) será substituído por Não e o evento *B* por Forte e Alta, portanto:

$$P(Não \mid Forte, Alta) = [P(Forte \mid Não) * P(Alta \mid Não) * P(Não)] / P(Forte) * P(Alta)$$
$$P(Não \mid Forte, Alta) = (0,6 * 0,8 * 0,36) / (0,43 * 0,5) = 0,80$$

O resultado final do algoritmo *Naive Bayes* é dado pela maior probabilidade encontrada através do Teorema de *Bayes.* Para a hipótese de que houve jogo de tênis a probabilidade resultou em 0,32 e para a hipótese de que não houve jogo de tênis o resultado da probabilidade foi 0,80. Neste caso, a maior probabilidade encontrada foi 0,80. Portanto, o exemplo sem rótulo receberá a classe Não. Logo, não houve jogo de tênis neste dia.

2.1.1.3 J48

O classificador J48, também conhecido como árvore de decisão C4.5, constrói uma árvore binária para fazer a classificação de entradas desconhecidas (PATIL; SHEREKAR, 2013).

Uma árvore de decisão é composta por vários nós que representam os atributos, por ramos que são ligações entre os nós que representam os valores dos atributos e nós folha que caracterizam as classes existentes em um problema de classificação (OLIVEIRA, 2013).

A construção da árvore de decisão ocorre de modo que a cada nó, o algoritmo escolhe o atributo que melhor subdivide o problema em subconjuntos de

acordo com sua classe (GIASSON *et al*, 2013). Para isso, o algoritmo de indução faz uma busca gulosa, calculando a entropia dos dados (OLIVEIRA, 2013).

A entropia representa a homogeneidade dos dados em relação a sua classificação. Se todos os elementos de um conjunto são membros da mesma classe, a entropia é mínima, ou seja, é igual a zero. Por outro lado, a entropia é máxima, ou seja, é igual a um quando representa um conjunto de dados heterogêneo. (MITCHELL, 1997). O cálculo da entropia para um conjunto *S* de entrada é dado pela equação (4).

$$E(S) = - p_i \log_2 p_i - p_j \log_2 p_j \quad (4)$$

Onde:

p_i é a proporção de dados em S que pertence a classe *i*;

p_j é a proporção de dados em S que pertence a classe *j*.

A medida para selecionar o melhor atributo para fazer a partição do conjunto de dados é o ganho de informação. Esta medida resulta na quantidade de diminuição da entropia quando determinado atributo for selecionado. Para isso, considere que *A* é um atributo pertencente ao conjunto *S*, *Valores(A)* é o conjunto de todos os possíveis valores de *A* e S_x é um subconjunto de *S* para o qual o atributo *A* tem valor igual a *x*. O cálculo do ganho de informação para o atributo *A* pode ser feito através da equação (5) (MITCHELL, 1997).

$$\text{Ganho}(S, A) = \text{Entropia}(S) - \sum_{x \in Valores(A)} (|S_x| / |S|)\ \text{Entropia}(S_x) \quad (5)$$

Para exemplificar a execução do algoritmo J48, suponha que *S* é um conjunto de treinamento referente ao problema de Jogo de Tênis, apresentado no Quadro 1. O conjunto *S* possui 14 exemplos, dentre os quais 9 pertencem a classe Sim e 5 a classe Não. Desses 14 exemplos, 6 dos positivos e 2 dos negativos possuem Vento igual a Fraco e, 3 dos positivos e 3 dos negativos apresentam Vento igual a Forte. O ganho de informação para construir a árvore de decisão do conjunto *S* através da escolha do atributo Vento como o melhor pode ser calculado como (MITCHELL, 1997):

$$Valores(Vento) = \{Fraco, Forte\}$$

$$S = [9+, 5-]$$

$$S_{Fraco} = [6+, 2-]$$

$$S_{Forte} = [3+, 3-]$$

$$\text{Entropia}(S) = - p_i \log_2 p_i - p_j \log_2 p_j$$

$$\text{Entropia}(S) = - (9/14) \log_2 (9/14) - (5/14) \log_2 (5/14) = 0{,}940$$

$$\text{Entropia}(S_{Fraco}) = - (6/8) \log_2 (6/8) - (2/8) \log_2 (2/8) = 0{,}811$$

$$\text{Entropia}(S_{Forte}) = - (3/8) \log_2 (3/8) - (3/8) \log_2 (3/8) = 1{,}000$$

$$\text{Ganho}\ (S, A) = \text{Entropia}(S) - \sum_{x \in Valores(A)} (|S_x| / |S|)\ \text{Entropia}(S_x)$$

$$\text{Ganho}\ (S, \text{Vento}) = \text{Entropia}(S) - \sum_{x \in (Fraco,\ Forte)} (|S_x| / |S|)\ \text{Entropia}(S_x)$$

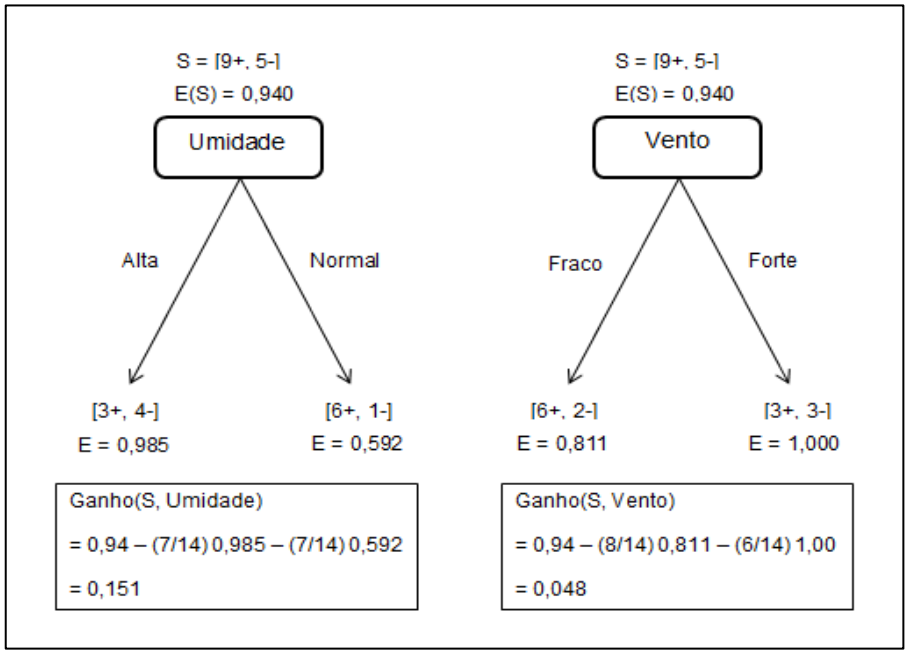
$$\text{Ganho}(S, \text{Vento}) = \text{Entropia}(S) - (8/14)\text{Entropia}(S_{Fraco}) - (6/14)\text{Entropia}(S_{Forte})$$

$$\text{Ganho}(S, \text{Vento}) = 0{,}940 - (8/14) * 0{,}811 - (6/14) * 1{,}000 = 0{,}048$$

O resultado do ganho de informação para construir a árvore de decisão do conjunto *S* através da escolha de Vento como melhor atributo é 0,048. O cálculo para encontrar o ganho de informação pela escolha de outros atributos segue os mesmos passos. A Figura 5 apresenta o processo de seleção de melhor atributo entre os atributos Umidade e Vento para o conjunto *S* do problema Jogo de Tênis apresentado no Quadro 1.

Na Figura 5, *S* possui 9 exemplos positivos e 5 negativos representado por *S* = [9+, 5-]. E(*S*) caracteriza o valor de entropia para o conjunto *S*. Há duas opções para melhor atributo: Umidade e Vento. Do conjunto *S* com 14 exemplos, quando a Umidade é Alta tem-se 3 exemplos positivos e 4 negativos, já quando a Umidade é Normal há 6 exemplos positivos e 1 negativo. Seguindo a mesma ideia para o atributo Vento, quando o mesmo é Fraco tem-se 6 exemplos positivos e 2 negativos e, quando é Forte há 3 exemplos positivos e 3 negativos. Tais quantidades são utilizadas para calcular a Entropia E de cada valor dos atributos. Por fim, é calculado o ganho de informação para Umidade e Vento. Nesse caso, o melhor atributo é a Umidade, pois seu ganho de informação é maior que o do Vento.

Figura 5 - Processo de seleção do melhor atributo para construção de árvore de decisão



Fonte: Mitchell (1997)

Para classificar um exemplo desconhecido é necessário seguir os nós sequentes da árvore de decisão, de acordo com os valores dos atributos recebidos, até chegar a um nó folha que representa a classe que será atribuída.

2.1.2 Avaliação de Classificadores Monorrótulo

Algoritmos de classificação monorrótulo podem ser avaliados a partir de uma matriz de confusão (MC), também chamada de matriz de contingência. Essa matriz possui duas dimensões com valores referentes à quantidade de exemplos preditos corretamente e incorretamente para cada classe (VILLANI, 2013).

O Quadro 3 mostra uma matriz de confusão de um classificador que possui duas classes: Positiva e Negativa. Nesta matriz de confusão, os significados das siglas VP, FN, FP e VN são (CERRI, 2010):

- VP (Verdadeiros positivos) - quantidade de exemplos preditos corretamente na classe positiva;
- FN (Falsos negativos) - quantidade de exemplos preditos com a classe negativa, mas que pertencem a classe positiva;

- FP (Falsos positivos) - quantidade de exemplos preditos com a classe positiva, mas que pertencem a classe negativa;
- VN (Verdadeiros negativos) - quantidade de exemplos preditos corretamente com a classe negativa.

Quadro 3 - Matriz de Confusão

		Classe Predita	
		Positiva	**Negativa**
Classe Verdadeira	**Positiva**	VP	FN
	Negativa	FP	VN

Fonte: Autoria própria

Com as informações obtidas na matriz de confusão é possível encontrar a taxa de acerto (TA), também conhecida como acurácia (ACC) e a taxa de erro (TE) de um classificador, através das equações (6) e (7) respectivamente (BORGES, 2012).

$$TA = \frac{|VN| + |VP|}{|VN| + |VP| + |FP| + |FN|} \tag{6}$$

$$TE = \frac{|FN| + |FP|}{|VN| + |VP| + |FP| + |FN|} \text{ ou } TE = 1 - TA \tag{7}$$

Para avaliar o desempenho de um classificador para cada rótulo existente em um problema de classificação foram criadas as medidas de revocação (R) e precisão (P).

A revocação, também conhecida como sensibilidade ou VP, representada na equação (8), calcula a probabilidade de um exemplo que pertence à classe positiva, ser classificado como positivo (CERRI, 2010). Em outras palavras, a revocação verifica se todos os rótulos verdadeiros dos exemplos foram preditos, se sim o resultado da medida valerá 1, caso contrário o valor da medida será penalizado de acordo com a quantidade de rótulos que deveriam ser preditos, mas não foram. A revocação não considera os rótulos preditos incorretamente (SILVA, 2014).

$$R = \frac{|VP|}{|VP| + |FN|} \tag{8}$$

A precisão mostrada na equação (9) é considerada uma medida que calcula a probabilidade de uma predição positiva estar correta em relação a todos os exemplos fornecidos ao problema de classificação (METZ, 2011). Em outras palavras, a precisão verifica se as classes preditas são verdadeiras e não leva em consideração as classes que deveriam ter sido preditas (SILVA, 2014).

$$P = \frac{|VP|}{|VP| + |FP|} \tag{9}$$

A combinação das medidas de precisão e revocação dá origem a uma nova medida, denominada de medida F que se trata de uma média harmônica ponderada das medidas combinadas. Na medida F, β é uma constante que define um peso de importância para as medidas precisão e revocação, na qual aumentando o valor de β, aumenta-se a importância da revocação e diminuindo o valor de β, aumenta-se o peso da precisão (VILLANI, 2013). A medida F é calculada através da equação (10).

$$\text{Medida F} = \frac{(\beta^2 + 1) * P * R}{\beta^2 * P + R} \tag{10}$$

As medidas de revocação e precisão analisam aspectos diferentes, com a combinação dessas medidas através da medida F é possível unir as vantagens de ambas. O resultado da medida F é um valor [0,1] e quanto mais próximo de 1, melhor é o desempenho do classificador.

2.2 CLASSIFICAÇÃO MULTIRRÓTULO

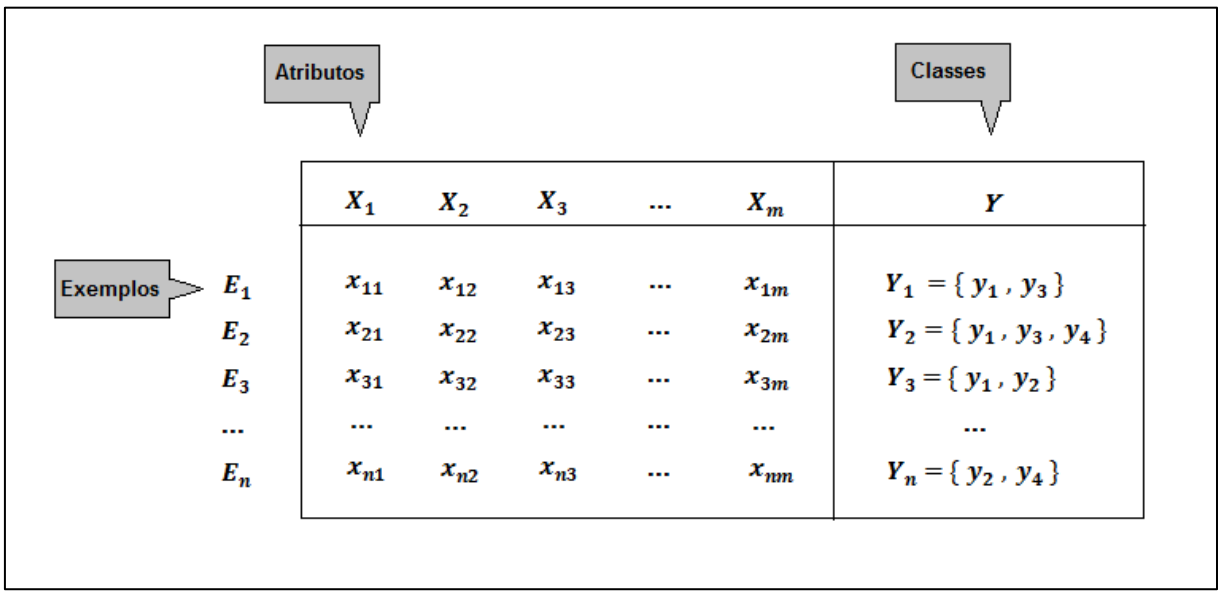
A maioria dos algoritmos de classificação considera que cada um dos exemplos fornecidos ao algoritmo de treinamento está associado a somente um rótulo (classe) e então gera um classificador capaz de atribuir uma única classe para

novos exemplos pertencentes ao conjunto de teste. No entanto, existem domínios em que um exemplo pode ser associado a mais de um rótulo simultaneamente, esse processo é denominado de classificação multirrótulo (METZ, 2011).

Na classificação multirrótulo, a entrada dos dados consiste em um conjunto de *N* exemplos. Cada exemplo E_n pode ser associado a mais de um rótulo ao mesmo tempo, ou seja, a um conjunto de rótulos Y_n, onde $Y_n \subset TY$, sendo *TY* o conjunto que contêm todos os rótulos existentes em um determinado problema de classificação (COELHO; ESMIN; JÚNIOR, 2011).

Um classificador multirrótulo pode ser representado pela função H: $E_n \rightarrow 2^{TY}$, onde E_n é um exemplo associado por um classificador *H*, de acordo com seus atributos X_m, a um conjunto de rótulos Y_n pertencente a 2^{TY} que é o conjunto com todas as possíveis combinações de rótulos para um determinado exemplo participante de um processo de classificação multirrótulo (CERRI, 2010). A representação dos exemplos multirrótulos pode ser visualizada na Figura 6, onde são demonstrados os conceitos de exemplos, atributos e classes.

Figura 6 - Representação de um conjunto de dados multirrótulo



Fonte: Autoria própria

Na Figura 6, exemplos são representados por linhas, ou seja, há *n* exemplos neste conjunto. Cada atributo é representado por uma coluna. Neste conjunto, existem *m* atributos que descrevem as características de cada exemplo. Por fim, existe uma coluna especial que representa as classes. Como este conjunto se refere

a um problema multirrótulo, cada exemplo pode estar associado a um conjunto de classes, ou seja, a mais de uma classe ao mesmo tempo.

Quadro 4 - Exemplo de um conjunto de dados multirrótulo

Exemplo	Classes
Filme1	{Comédia, Ação}
Filme2	{Terror}
Filme3	{Terror, Comédia}
Filme4	{Comédia, Romance, Ação}

Fonte: Rodrigues (2014)

Para exemplificar o processo de classificação multirrótulo, o Quadro 4 mostra um conjunto de dados com quatro exemplos de filmes, onde cada filme está associado a uma ou mais classes ao mesmo tempo. Cada filme pode ter no máximo quatro rótulos, representados pelas categorias: terror, comédia, romance e ação.

2.2.1 Métodos de Classificação Multirrótulo

Na literatura há diversas formas de tratar problemas de classificação multirrótulo, uma maneira comum consiste em transformar um problema multirrótulo em vários problemas de classificação monorrótulo. Com isso, classificadores monorrótulo podem ser combinados a fim de encontrar soluções para problemas de classificação multirrótulo (SANTOS, 2012).

Quadro 5 - Problema multirrótulo

Exemplo	Classes
Exemplo1	$\{y_2, y_4\}$
Exemplo2	$\{y_1\}$
Exemplo3	$\{y_1, y_2\}$
Exemplo4	$\{y_2, y_3, y_4\}$

Fonte: Autoria própria

O Quadro 5 apresenta um conjunto de exemplos multirrótulo que representam um problema de classificação multirrótulo, no qual um exemplo pode estar associado a mais de um rótulo ao mesmo tempo. Neste capítulo serão apresentados três métodos que podem ser utilizados para resolver este problema.

2.2.1.1 *Binary Relevance*

No método *Binary Relevance* (BR), os exemplos multirrótulo, ou seja, exemplos que estão associados a mais de um rótulo simultaneamente são transformados em N conjuntos de dados S_n, onde cada S_n, é referente a um rótulo y_i, $i = 1, ..., N$.

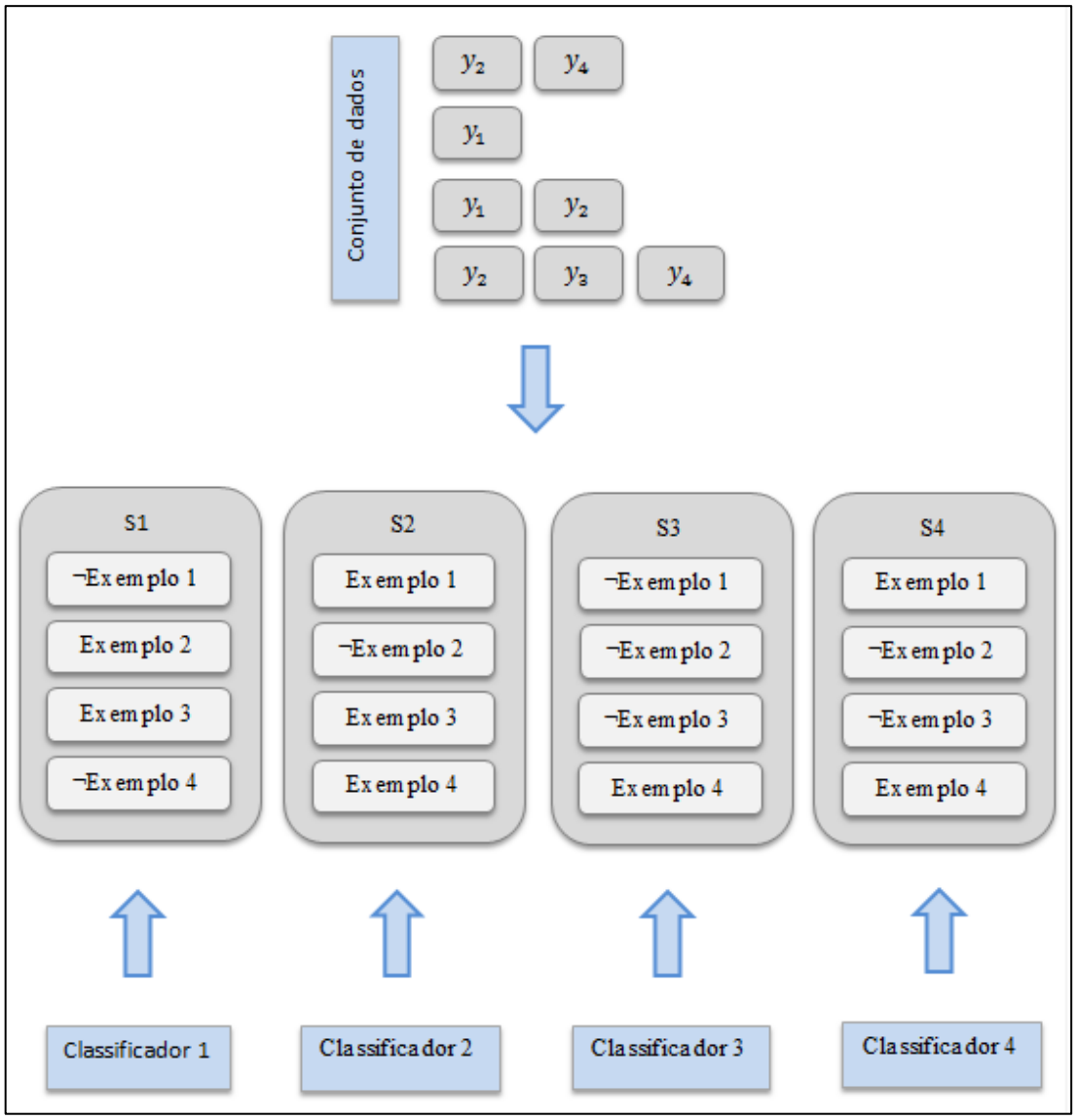
O método BR transforma problemas multirrótulo em vários problemas monorrótulo binários, para isso cada exemplo do conjunto de dados fornecido deve ser classificado como pertencente ao conjunto S_n, quando o mesmo está associado ao rótulo y_i correspondente, ou não pertencente (¬), caso contrário (SANTOS, 2012).

Após ter vários problemas monorrótulo, é construído um classificador para cada conjunto de dados S_n através de um mesmo algoritmo de aprendizagem para classificação monorrótulo. Para classificar um novo exemplo, o mesmo deve ser fornecido para todos os classificadores S_n referentes aos rótulos y_i que o exemplo original possui. Se o classificador predizer que o exemplo é positivo, então a predição está correta, logo o rótulo ao qual o classificador se refere é adicionado ao conjunto solução (GAMA; BERNARDINI; ZADROZNY, 2013).

A vantagem desse método é a simplicidade em como é construído o classificador e como é feita a classificação e a desvantagem é que ele desconsidera a relação entre os rótulos na construção do classificador multirrótulo (CHERMAN; METZ; MONARD, 2010).

A Figura 7 ilustra a transformação do problema multirrótulo mostrado no Quadro 5 em diversos problemas monorrótulo, e em seguida a construção de classificadores BR.

Figura 7 - Processo de construção de classificadores BR



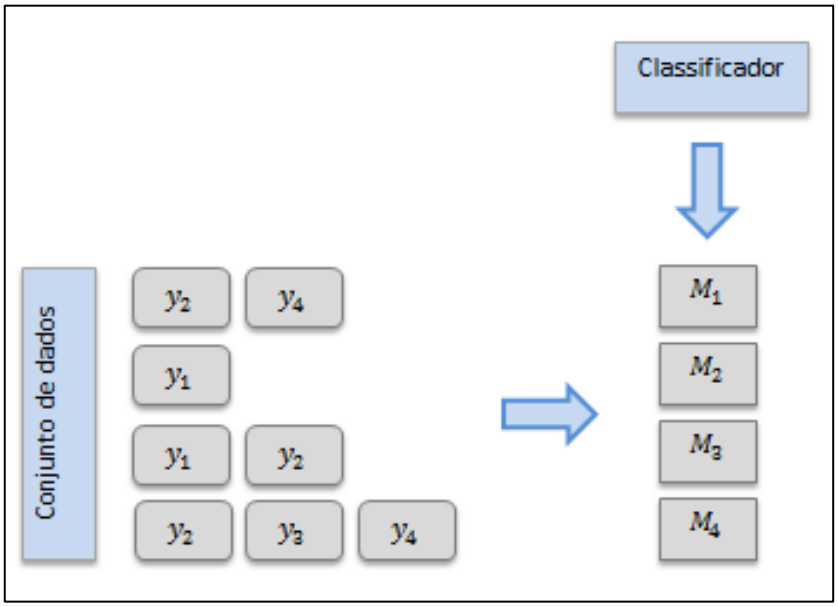
Fonte: Adaptado de Santos (2012)

Como pode-se observar na Figura 7, os exemplos do conjunto de dados estão associados a quatro rótulos: y_1, y_2, y_3 e y_4. Como existem ao todo quatro rótulos, então são criados quatro novos conjuntos de problemas monorrótulo: S_1 referente ao rótulo y_1, S_2 relativo ao rótulo y_2, S_3 correspondente ao rótulo y_3 e S_4 que se refere ao rótulo y_4. Cada exemplo do conjunto de dados deve ser classificado como pertencente ou não ao conjunto S_n. O Exemplo 1 possui os rótulos y_2 e y_4, por isso é classificado como pertencente aos conjuntos S_2 e S_4 e não pertencente aos conjuntos S_1 e S_3. Todos os exemplos são classificados da mesma maneira. Após os conjuntos S_n estarem formados, é construído um classificador para cada conjunto S_n que definirá se um exemplo desconhecido pertence ou não ao rótulo y_i.

2.2.1.2 *Label Powerset*

O método *Label Powerset* (LP) transforma um problema multirrótulo em vários problemas monorrótulo. Este método verifica na etapa de treinamento todos os conjuntos únicos de rótulos associados aos exemplos e os transforma em valores unitários para o atributo classe, a partir dos quais um classificador é construído. Por exemplo, o dado multirrótulo {y_2, y_4}, poderia ser transformado em um índice representado pela letra "M_1", assim cada vez que o índice "M_1" for utilizado, automaticamente são recuperados os rótulos y_2 e y_4 (METZ, 2011). O processo de criação de um classificador LP para o problema multirrótulo apresentado no Quadro 5 é representado pela Figura 8.

Figura 8 - Processo de construção de um classificador LP



Fonte: Adaptado de Santos (2012)

O LP considera a relação entre os rótulos na construção do classificador e essa é sua vantagem em relação ao método BR. A desvantagem desse método é que pode ocorrer um crescimento exponencial de subconjuntos de rótulos, resultando em muitas classes com poucos exemplos associados, o que aumenta o custo computacional para a execução do LP. Outra desvantagem é que o LP pode prever somente os conjuntos de rótulos que fizeram parte da etapa de treinamento, o que o torna limitado, já que novos conjuntos poderão aparecer no conjunto de teste (SANTOS, 2012).

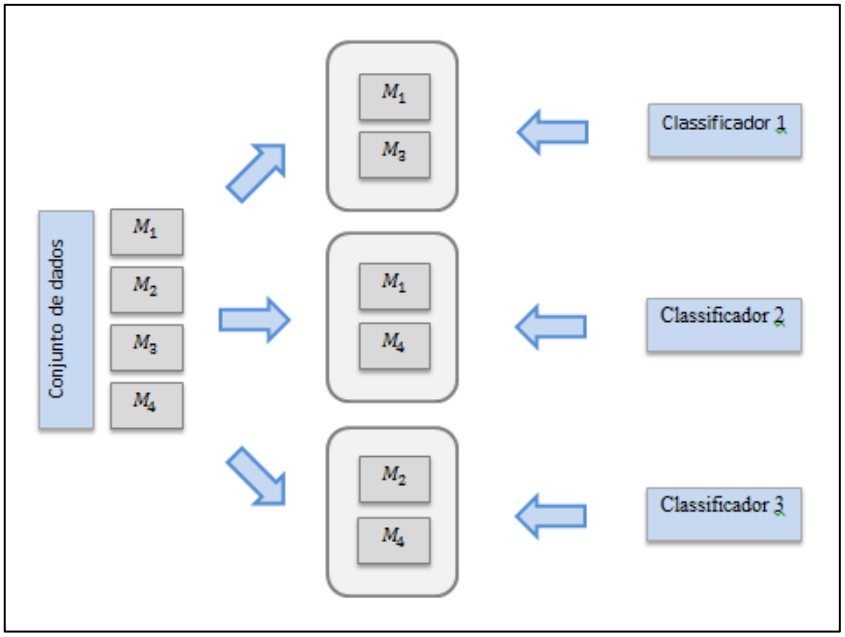
2.2.1.3 *Random k-labelsets*

Este método foi proposto em Tsoumakas, Katakis e Vlahavas (2010) com o objetivo de diminuir os problemas do LP, de modo a considerar as relações entre os rótulos e reduzir o problema da ocorrência de muitas classes com poucos exemplos associados.

No método de classificação *Random k-labelsets* (RAKEL) é construído um comitê de classificadores LP, onde *k* representa o tamanho do *labelset*. Cada classificador do comitê de classificadores LP é treinado utilizando um subconjunto aleatório de *labelsets* (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010). O processo de construção de classificadores RAKEL pode ser visualizado na Figura 9.

É possível afirmar que no RAKEL os classificadores monorrótulo consideram as correlações entre os rótulos e também são aplicados em subtarefas que possuem quantidades adequadas de rótulos e exemplos por rótulos (RODRIGUES, 2014).

Figura 9 - Processo de construção de classificadores RAKEL



Fonte: Adaptado de Santos (2012)

Para classificar um novo exemplo, cada classificador gera uma decisão binária para cada rótulo existente no seu *labelset* Y_i, ou seja, para cada rótulo em Y_i, o classificador responde se o rótulo do exemplo pertence ou não ao conjunto de rótulos solução (SILVA, 2014).

2.2.2 Avaliação de Classificadores Multirrótulo

A dificuldade de avaliar classificadores multirrótulo é maior do que classificadores monorrótulo, pois em um problema monorrótulo cada exemplo tem apenas um rótulo predito, portanto há duas possibilidades de resultado: o rótulo atribuído é igual a saída esperada ou diferente. Já na classificação multirrótulo um exemplo pode estar associado a um conjunto de rótulos, no qual a saída não seja igual ao conjunto esperado, mas pode estar classificada de maneira parcialmente correta (SILVA, 2014).

Uma solução dada a um problema multirrótulo é considerada parcialmente correta se o conjunto de rótulos atribuídos a um exemplo está correto, porém faltou atribuir um ou mais rótulos neste conjunto ou quando rótulos são atribuídos indevidamente ao conjunto solução.

A avaliação de classificadores multirrótulo requer medidas adequadas para o problema multirrótulo, no entanto, medidas de avaliação utilizadas em classificadores monorrótulo podem ser utilizadas como base neste processo (VILLANI, 2013).

Um exemplo multirrótulo é formado por um conjunto de exemplos monorrótulo. Devido ao fato de um exemplo se tratar de um conjunto então operações de conjuntos podem ser utilizadas no processo de avaliação das predições feitas por classificadores multirrótulo. Operações como intersecção e união permitem fazer a comparação entre o conjunto de rótulos esperados e o conjunto de rótulos preditos para um exemplo (METZ, 2011).

As medidas acurácia, precisão e revocação originais, ou seja, medidas utilizadas para avaliar predições feitas por classificadores monorrótulo foram adaptadas para conseguirem fazer a avaliação de predições feitas por classificadores multirrótulo e por isso cada medida pode ser utilizada com os mesmos objetivos (RIVOLLI; CARVALHO, 2015).

As medidas acurácia, precisão e revocação podem ser calculadas para um conjunto de N exemplos multirrótulo através das equações (11), (12) e (13), onde Y_i e Z_i representam respectivamente o conjunto de rótulos esperados e o conjunto de rótulos associados para um determinado exemplo (VALLIM, 2009).

$$\text{ACC} = \frac{1}{N} \sum_{i=1}^{N} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{11}$$

$$\text{P} = \frac{1}{N} \sum_{i=1}^{N} \frac{|Y_i \cap Z_i|}{|Z_i|} \tag{12}$$

$$\text{R} = \frac{1}{N} \sum_{i=1}^{N} \frac{|Y_i \cap Z_i|}{|Y_i|} \tag{13}$$

A medida F também foi adaptada para oferecer em uma única medida as vantagens e desvantagens da utilização das medidas precisão e revocação para *N* exemplos multirrótulos. A medida F é calculada através da equação (14), onde Y_i representa o conjunto de rótulos esperados para um exemplo e Z_i é o seu conjunto de rótulos preditos.

$$\text{F} = \frac{1}{N} \sum_{i=1}^{N} \frac{2 * |Z_i \cap Y_i|}{|Z_i| + |Y_i|} \tag{14}$$

A medida *Hamming Loss* (HL), representada pela equação (15), calcula a média percentual de predições feitas incorretamente em relação a quantidade total de rótulos existentes no problema (*q*). Na equação, a função XOR(Z_i, Y_i) representa a função OR exclusivo, ou seja, o resultado contêm os rótulos que fazem parte somente de um dos dois conjuntos. Os conjuntos Y_i e Z_i representam respectivamente os rótulos esperados e os rótulos associados para um determinado exemplo (SILVA, 2014).

$$\text{HL} = \frac{1}{N} \sum_{i=1}^{N} \frac{|\text{XOR}(Z_i, Y_i)|}{|q|} \tag{15}$$

A medida de avaliação HL leva em consideração a perda, por isso zero é o seu valor ideal, ou seja, quanto mais próximo de zero for o resultado, melhor é o classificador.

2.3 CARACTERÍSTICAS DOS DADOS

Ao trabalhar com problemas de classificação multirrótulo é importante conhecer as características dos dados que serão utilizados no processo. Há dois tipos de dados: quantitativos e qualitativos. Os quantitativos são descritos por valores numéricos, já os qualitativos são representados por valores nominais (OLIVEIRA, 2009).

Antes de aplicar algoritmos de classificação multirrótulo é necessário preparar os dados. Valores qualitativos devem ser transformados em valores quantitativos (BATISTA, 2003). Essa transformação é feita através de uma normalização que mapeia os valores qualitativos para números, geralmente com faixas de valores pequenas, como de 0 a 1 (OLIVEIRA, 2009). Para exemplificar, os atributos qualitativos sim e não, poderiam ser transformados respectivamente em 0 e 1.

Após transformar todos os dados necessários, ou seja, possuir um conjunto somente de dados quantitativos pode-se explorá-lo. No entanto, podem existir dados que comprometam a qualidade dos algoritmos de classificação, como: valores nulos e desconhecidos.

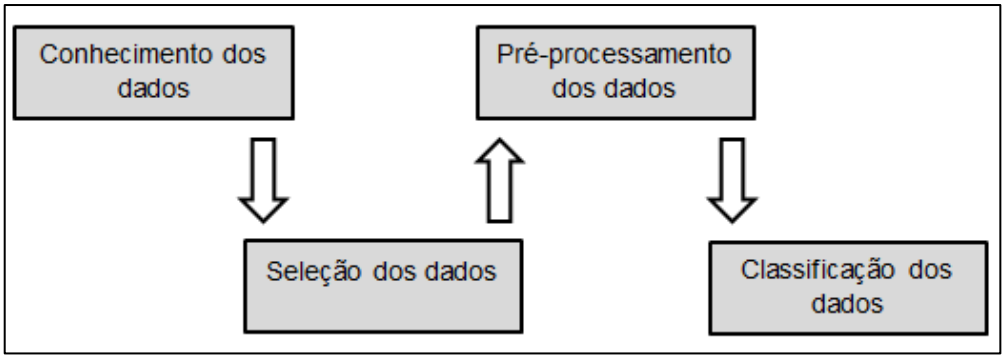
A ausência de valores em bases de dados utilizadas por algoritmos de classificação podem afetar o desempenho do algoritmo e causar resultados incorretos, dificultando possíveis análises. Além disso, existem algoritmos que não podem ser aplicados em bases de dados com valores faltantes. O ideal é que antes de ocorrer o processo de classificação, ocorra o pré-processamento dos dados para que problemas com bases de dados, como valores corrompidos, desconhecidos e incompletos sejam resolvidos (BATISTA, 2003).

No processo de classificação existe a fase de pré-processamento dos dados que tem por objetivo melhorar os valores, identificando e tratando problemas presentes em bases de dados antes que algoritmos de classificação sejam executados (BATISTA, 2003).

Na Figura 10 é possível ver as fases do processo de classificação de dados. Inicialmente deve haver o conhecimento dos dados, ou seja, é importante identificar a estrutura das bases de dados, os tipos dos dados e quais atributos farão parte da classificação. Em seguida, todos os dados necessários para a classificação deverão

ser selecionados. Depois disso, deve-se fazer a etapa de pré-processamento dos dados buscando aprimorar a qualidade dos mesmos. Somente após ocorrer o pré-processamento dos dados é que acontece a classificação dos dados.

Figura 10 - Fases do processo de classificação de dados



Fonte: Autoria própria

A fase de pré-processamento dos dados envolve grandes quantidades de dados e muitos desses são desconhecidos. Para resolver este problema, foram desenvolvidas técnicas que substituem dados faltantes por estimativas de valores, sendo a imputação de valores faltantes a mais comum (NUNES, 2007). Na seção 2.4 serão apresentadas as principais técnicas de imputação.

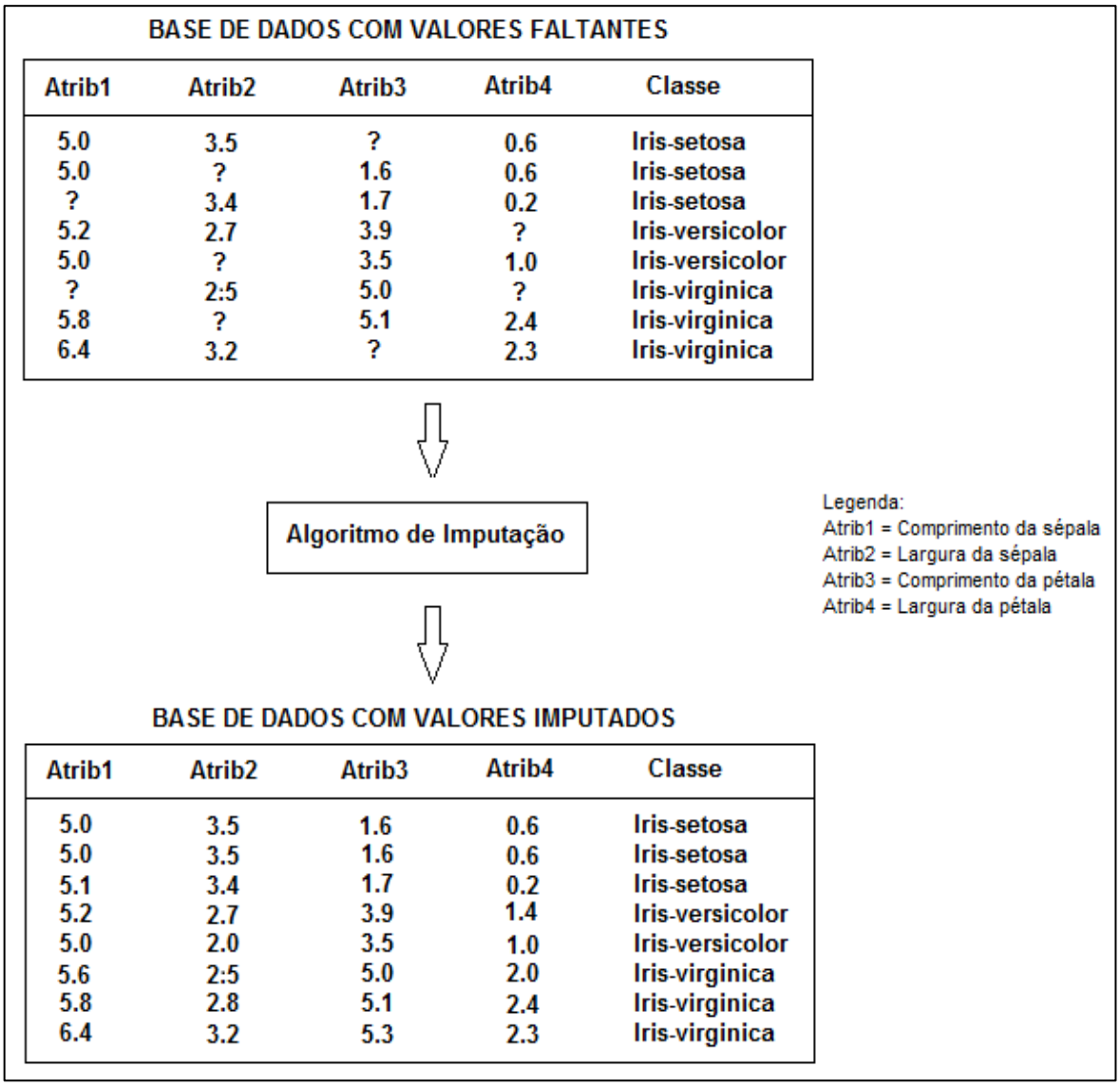
2.4 IMPUTAÇÃO DE VALORES FALTANTES

A técnica de imputação de dados tem sido bastante utilizada para resolver o problema de valores faltantes. Imputações são procedimentos de tratamento de dados que substituem os valores omissos de uma base de dados por valores específicos individuais (DIAS, ALBIERI, 1992).

Ao realizar o processo de imputação deve-se ter cuidado para não distorcer os dados ou causar a discrepância das estimativas, pois se isto acontecer o resultado pode ser afetado, já que os valores substituídos são diferentes do conjunto de dados inicial (BRITTO, 2005).

A Figura 11 mostra como funciona o processo de imputação em uma base de dados com valores faltantes. Na literatura, valores incompletos de atributos de bases de dados são representados por um símbolo de interrogação (?).

Figura 11 - Imputação de dados



Fonte: Adaptado de Borges (2012)

Na Figura 11, inicialmente, o conjunto de exemplos com valores faltantes é fornecido para um algoritmo de imputação para que ocorra o tratamento dos dados. O algoritmo de imputação calcula valores a partir dos dados já existentes na base de dados fornecida e os insere no lugar dos valores desconhecidos. Após ocorrer o processo de imputação sobre os dados, os valores incompletos são substituídos por valores estimados mais próximos aos reais, tornando a base de dados completa. Neste caso, os valores imputados são apenas ilustrativos, pois existem várias técnicas de imputação e cada uma pode resultar em valores diferentes.

A imputação de dados visa tornar uma base de dados com valores omissos em uma base de dados completa. Existem diversas técnicas de imputação que vão

desde as mais simples, como a média, até as mais complexas que utilizam aprendizagem de máquina (GHINOZZI, 2012).

Quadro 6 - Base de dados com valores faltantes

Atrib1	Atrib2	Atrib3	Atrib4
5,0	3,5	?	0,6
5,0	?	1,7	0,6
?	3,5	1,7	0,2
5,2	2,7	3,9	?
5,1	?	3,5	1,0

Fonte: Autoria própria

O Quadro 6 representa uma base de dados com valores faltantes, na qual pode ser utilizado o processo de imputação para completar o conjunto. Neste capítulo serão apresentadas quatro técnicas de imputação para resolver este problema.

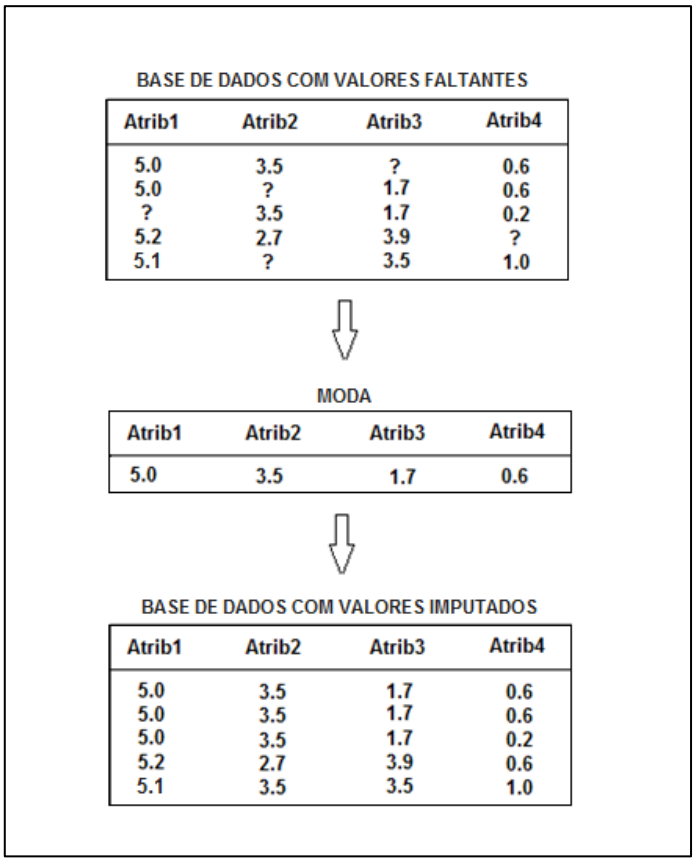
2.4.1 Imputação pela Moda

Na imputação pela Moda, também conhecida como Imputação pelo Valor Mais Comum, os dados ausentes devem ser substituídos pelo valor que mais aparece no atributo de interesse, considerando todos os exemplos do conjunto de dados (OLIVEIRA, 2009).

Esta técnica possui um bom desempenho computacional, porém não leva em consideração o relacionamento entre os atributos podendo gerar resultados bem diferentes dos valores reais.

A Figura 12 mostra a aplicação da Imputação pela Moda para resolver o problema da base de dados com valores faltantes apresentada no Quadro 6.

Figura 12 - Imputação pela Moda



Fonte: Autoria própria

A Figura 12 mostra que para realizar a Imputação pela Moda deve-se verificar o valor que mais se repete para cada atributo. Em seguida, todos os valores faltantes de um determinado atributo são substituídos pela moda encontrada. Por exemplo, na primeira coluna da base de dados com valores faltantes, o valor que mais se repete é 5.0, por isso 5.0 é a moda para o Atrib1. Então, todos os valores faltantes de Atrib1 foram substituídos por 5.0. Os valores incompletos dos demais atributos foram imputados utilizando os mesmos passos.

2.4.2 Imputação pela Média

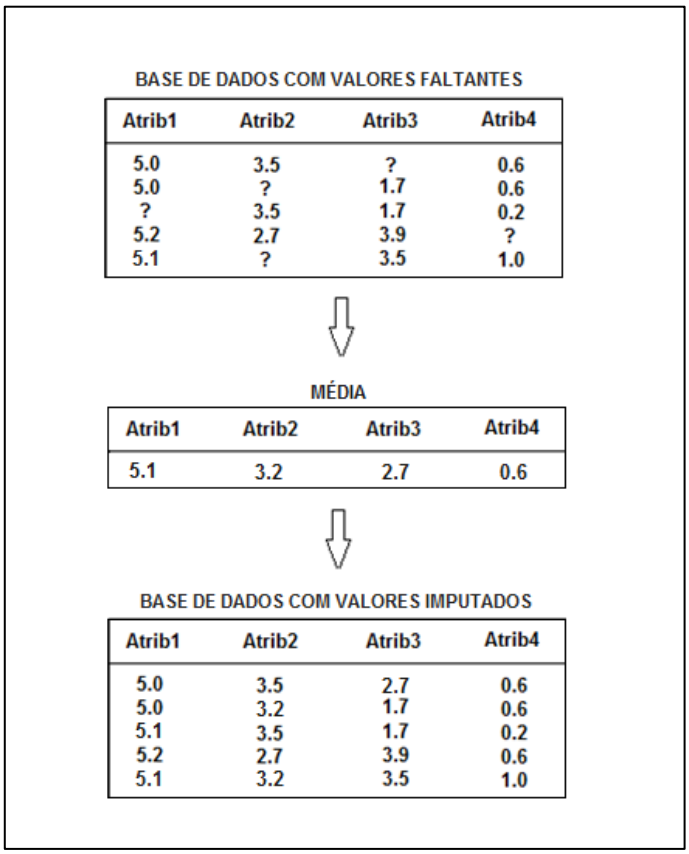
Técnica antiga e frequentemente utilizada, na qual um dado faltante é substituído por um valor calculado pela média do atributo em questão de todos os exemplos do conjunto de dados (CASTRO, 2014).

A principal vantagem da imputação pela média é a implementação simples e a principal desvantagem é que comprime os dados observados, já que todos os

dados com o mesmo atributo perdido terão um único valor constante: a média (WILSON, 2010; NUNES; KLÜCK; FACHEL, 2009). Outro problema é que a média pode ser influenciada por valores extremos, podendo gerar valores divergentes dos esperados.

A Figura 13 mostra a aplicação da Imputação pela Média para resolver o problema da base de dados com valores faltantes apresentada no Quadro 6.

Figura 13 - Imputação pela Média



Fonte: Autoria própria

A Figura 13 demonstra que para realizar a Imputação pela Média deve-se calcular a média dos valores válidos para cada atributo. Em seguida, todos os valores faltantes de um determinado atributo são substituídos pela média encontrada. Por exemplo, considere a primeira coluna da base de dados com valores faltantes que se refere ao Atrib1. A média para Atrib1 é dada por (5.0 + 5.0 + 5.2 + 5.1) / 4 que resulta no valor 5.1. Valores incompletos não são considerados no cálculo da média. Calculou-se a média para os demais atributos da mesma maneira. No final, todos os valores faltantes de cada atributo foram substituídos por suas respectivas médias.

2.4.3 Imputação pela Mediana

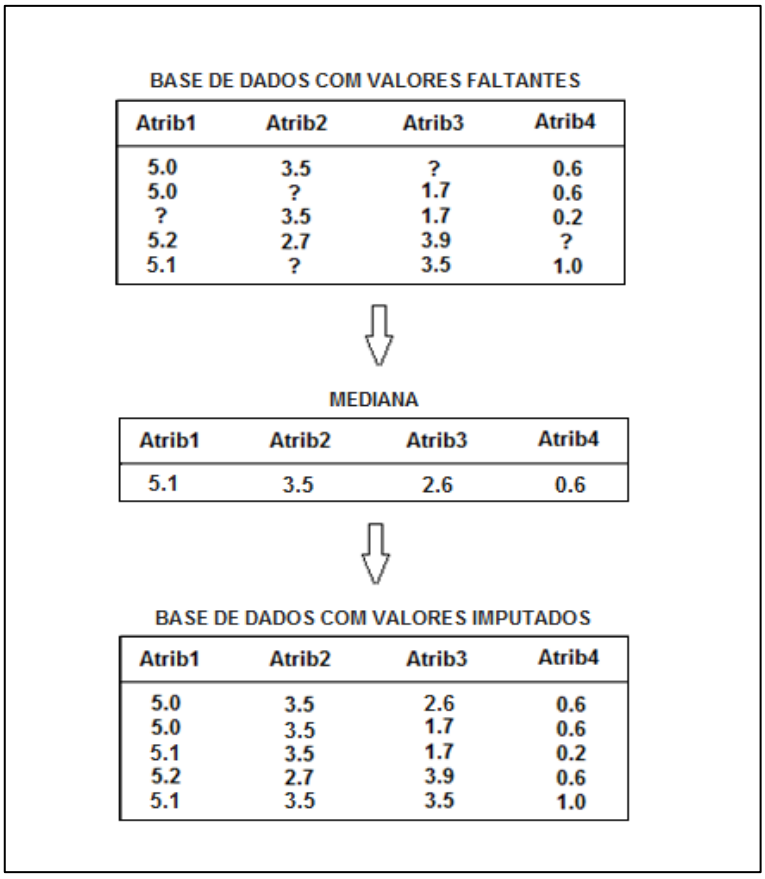
Técnica que substitui um dado faltante pela mediana do atributo calculada a partir dos valores do mesmo atributo de todos os exemplos existentes no conjunto de dados (NUNES, 2007).

A Imputação pela Mediana tem a vantagem de possibilitar uma implementação simples e possui a mesma desvantagem da média: todos os valores faltantes do mesmo atributo terão um valor constante: a mediana.

A vantagem da mediana em relação à média é que a mediana é menos sensível a valores extremos, sofrendo menor influência nos resultados se comparada com a média. Porém, nem sempre o valor da mediana consegue representar bem um conjunto de dados.

A Figura 14 mostra a aplicação da Imputação pela Mediana para resolver o problema da base de dados com valores faltantes apresentada no Quadro 6.

Figura 14 - Imputação pela Mediana



Fonte: Autoria própria

A Figura 14 demonstra que para realizar a Imputação pela Mediana deve-se calcular a mediana dos valores válidos para cada atributo. Em seguida, todos os valores faltantes de um determinado atributo são substituídos pela sua respectiva mediana. Por exemplo, considere a primeira coluna da base de dados com valores faltantes que se refere ao Atrib1. Para encontrar a mediana de Atrib1 colocam-se em ordem crescente os valores existentes: 5.0, 5.0, 5.1, 5.2. A mediana de Atrib1 é dada pelo número que divide o conjunto de dados ordenados em dois subconjuntos de mesmo tamanho, ou seja, é o valor do meio do conjunto. Como há um número par de elementos no conjunto, a mediana é dada pela média de 5.0 e 5.1 que resulta em 5.05, mas arredondando para apenas uma casa decimal equivale a 5.1. Valores incompletos não são considerados no cálculo da mediana.

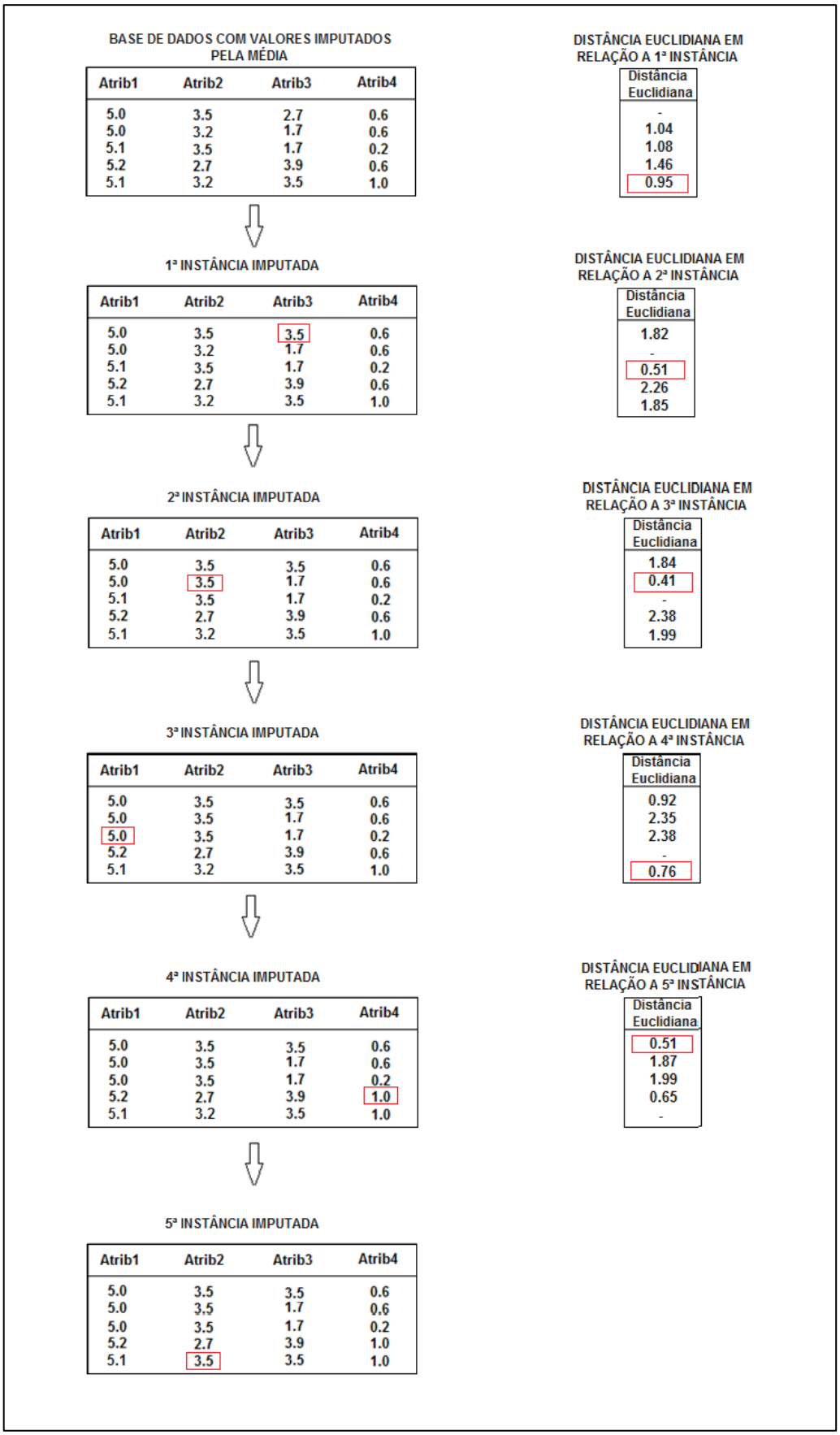
2.4.4 Imputação KNN Iterativo

O algoritmo de imputação KNN Iterativo se inicia com a substituição dos valores faltantes através da técnica da média para tornar a base de dados completa. Em seguida, é realizado o processo de imputação KNN de maneira iterativa, ou seja, em *n* ciclos (SILVA, 2010).

Em cada ciclo do KNN Iterativo, são buscadas as *k* instâncias do conjunto de dados mais próximas a um determinado exemplo que possui valores omissos. As instâncias mais próximas são calculadas através da Distância Euclidiana (equação 1). No fim de cada ciclo, os valores estimados para os atributos ausentes dessa instância são recalculados e substituídos pelas médias dos valores dos atributos das *k* instâncias mais próximas encontradas (SILVA, 2010).

A Figura 15 apresenta a execução do KNN Iterativo com um ciclo para resolver o problema da base de dados com valores faltantes apresentada no Quadro 6. Neste exemplo, considere que *k* é igual a um, por isso sempre será encontrada apenas uma instância mais próxima para realizar a imputação dos valores. Considere também que a base de dados já sofreu Imputação pela Média (passo inicial da execução do KNN Iterativo). As menores Distâncias Euclidianas encontradas e os valores imputados estão destacados.

Figura 15 - Imputação KNN Iterativo



Fonte: Autoria própria

Na Figura 15, o primeiro conjunto se refere a base de dados que sofreu Imputação pela Média. A partir dele foram calculadas as Distâncias Euclidianas de todas as instâncias em relação à primeira instância. O menor valor encontrado foi 0.95 que significa que a quinta instância é a que mais se aproximou da primeira. Portanto, os valores da primeira instância que estavam faltando na base de dados antes de realizar a Imputação pela Média (conforme pode ser visualizado no Quadro 6) foram substituídos pelos valores dos atributos equivalentes da quinta instância da base completa. O Atrib3 estava ausente, então foi substituído pelo valor de Atrib3 da quinta instância. Logo, Atrib3 da primeira instância recebeu o valor 3.5. Depois de imputar a primeira instância, imputaram-se as demais utilizando os novos conjuntos gerados para calcular as Distâncias Euclidianas.

2.5 TRABALHOS RELACIONADOS

O processo de imputação é o ponto principal deste trabalho, mas como a imputação é utilizada para preparar bases de dados que serão utilizadas por algoritmos de classificação, então torna-se necessário apresentar conceitos da tarefa de classificação, mais especificamente características de bases de dados multirrótulo, problemas de classificação multirrótulo, bem como suas formas de avaliação. São exemplos de trabalhos que citam problemas multirrótulo: Silva (2014), Rodrigues (2014), Villanni (2013), Borges (2012), Metz (2011), Tsoumakas *et al* (2010), Cerri (2010) e Vallim (2009).

Na literatura há vários trabalhos que apresentam tratamentos para bases de dados com valores omissos, sendo a imputação o mais utilizado. Existem diferentes métodos de imputação, desde métodos simples como a média até métodos que utilizam aprendizagem de máquina, como a imputação utilizando KNN. Os conceitos e os principais métodos de imputação para bases de dados com valores faltantes podem ser encontrados nos trabalhos de: Castro (2014), Ghinozzi (2012), Wilson (2010), Oliveira (2009), Nunes (2007) e Britto (2005).

Visando comparar o efeito de diferentes métodos de imputação Mundfrom e Whitcomb (1998) excluíram 11% dos pacientes de uma base de dados de um hospital para que a base ficasse com valores faltantes. A partir disso, foram

aplicados os métodos de imputação pela Média, Regressão e *Hot Deck* sobre os dados. A base de dados utilizada se trata de um problema monorrótulo, onde cada paciente pode receber somente um rótulo: “está doente ou não”.

No trabalho de Acuña e Rodriguez (2014) foram realizados testes para verificar a efetividade do tratamento de dados faltantes em bases de dados, utilizando três métodos de imputação: Média, Mediana e KNN. Para fazer este trabalho foram utilizadas doze bases de dados com diferentes porcentagens de valores ausentes: 1% a 20%. A imputação foi aplicada em todas as bases de dados e em seguida, foram aplicados classificadores monorrótulo em cada conjunto de dados para então calcular a acurácia de cada método. Todos os exemplos das bases de dados estão associados a somente um rótulo, portanto, se referem a problemas de classificação monorrótulo.

Batista e Monard (2003) analisaram métodos para tratamento de valores omissos em quatro bases de dados com porcentagens entre 1 e 20% de valores ausentes, nas quais foram aplicados os métodos de imputação utilizando KNN, Média e Moda. Em seguida, foram aplicados classificadores monorrótulo, para calcular a eficiência de cada método de imputação. Neste experimento todas as bases de dados são de problemas de classificação monorrótulo.

Diante dos trabalhos relacionados apresentados, pode-se observar que foram analisados algoritmos de imputação aplicados em bases de dados monorrótulo para que problemas de classificação monorrótulo pudessem ser solucionados. No entanto, ainda não foi estudada a eficiência de algoritmos de imputação aplicados em bases de dados multirrótulo, fator que será explorado neste trabalho.

3 EXPERIMENTOS

Este capítulo mostra os experimentos realizados e os resultados obtidos com os algoritmos de imputação utilizados: Imputação pela Moda, Imputação pela Média, Imputação pela Mediana e Imputação KNN Iterativo.

Para validar os resultados dos algoritmos de imputação foram usados os classificadores multirrótulo: BR, LP e RAKEL. Esses três algoritmos de classificação transformam um problema multirrótulo em vários problemas monorrótulo, por isso também foi necessário o uso dos classificadores monorrótulo: *Naive Bayes* e J48. As medidas Acurácia, HL e Medida F avaliaram os resultados dos classificadores.

Neste capítulo são apresentadas as bases de dados escolhidas, a metodologia utilizada com todas as etapas detalhadas, o *framework* usado para fazer as classificações e as avaliações dos resultados.

3.1 BASES DE DADOS MULTIRRÓTULO

Para a realização deste trabalho foram selecionadas seis bases de dados multirrótulo de diferentes domínios de aplicação, nas quais não há valores ausentes. As bases escolhidas são: *Emotions*, *Enron*, *Mediamill*, *Medical, Scene* e *Yeast.* O Quadro 7 apresenta a descrição das bases de dados selecionadas.

Quadro 7 - Descrição das bases de dados multirrótulo

Base de Dados	Descrição
Emotions	Base de dados relacionada com a classificação de músicas de acordo com as emoções que cada uma proporciona.
Enron	Base de dados pertencente ao domínio texto.
Mediamill	Base de dados que tem como objetivo traduzir pixel para texto para fazer a recuperação de imagens.
Medical	A *Medical* é constituída por documentos que descrevem históricos de sintomas e prognósticos que são utilizados para prever códigos seguros.
Scene	Base de dados relacionada com a indexação semântica de cenas.
Yeast	Base de dados que está relacionada com a classificação de proteínas.

Fonte: Autoria própria

A Tabela 1 apresenta um resumo sobre as características das bases de dados selecionadas, informando a quantidade de instâncias, atributos e rótulos presentes em cada uma.

Tabela 1 - Resumo das características das bases de dados multirrótulo

Base	Instâncias	Atributos	Rótulos
Emotions	592	71	6
Enron	1.702	1.001	53
Mediamill	43.907	120	101
Medical	978	1.449	45
Scene	2.407	294	6
Yeast	2.417	103	14

Fonte: Autoria própria

De acordo com a Tabela 1, a base de dados com maior número de instâncias é a *Mediamill* com 43.907 e a base de dados com menor quantidade de instâncias é a *Emotions* com 592. *Medical* possui o maior número de atributos sendo 1.449 e *Emotions* possui apenas 71. A base de dados que contém maior número de rótulos é a *Mediamill* com 101 e as que possuem menores quantidades de rótulos são as bases *Emotions* e *Scene* com somente 6.

As bases de dados escolhidas podem ser encontradas em http://meka.sourceforge.net/#datasets e http://mulan.sourceforge.net/datasets-mlc.html.

3.1.1 Formato ARFF

As bases de dados são apresentadas no formato ARFF. Um arquivo ARFF é formado basicamente pelas características de um conjunto de dados, lista de atributos que inclui todas as suas classes e atributos e, lista de dados que se refere a todas as instâncias existentes no conjunto. A Figura 16 traz um exemplo de arquivo ARFF referente a uma base de dados multirrótulo.

Figura 16 - Exemplo de arquivo no formato ARFF

```
1  @relation 'Nome_Base: -C 3'
2
3  @attribute classe1 {0,1}
4  @attribute classe2 {0,1}
5  @attribute classe3 {0,1}
6  @attribute x1 numeric
7  @attribute x2 numeric
8  @attribute x3 numeric
9  @attribute x4 numeric
10
11 @data
12 0,1,1,0.13,0.07,0.22,0.60
13 0,1,0,0.54,0.35,0.15,0.79
14 0,0,1,0.17,0.24,0.25,0.43
```

Fonte: Autoria Própria

Na Figura 16, a linha prefixada com *@relation* é referente as características da base de dados, onde Nome_Base é o nome do conjunto e 3 é a sua quantidade de classes. O número de classes é definido após -C. Em seguida, aparece a lista de atributos, especificados por *@attribute*, com seus respectivos nomes e tipos de dados. Na lista de atributos, inicialmente são apresentadas as classes e em seguida os atributos. Por fim, aparece a lista de dados (demarcada com *@data*) com todas as instâncias da base de dados. Cada valor de uma instância é referente a um atributo de acordo com a ordem em que se encontram. Neste caso, os três primeiros valores de uma instância são referentes às classes classe1, classe2 e classe3 e os demais valores aos atributos x1, x2, x3 e x4 respectivamente.

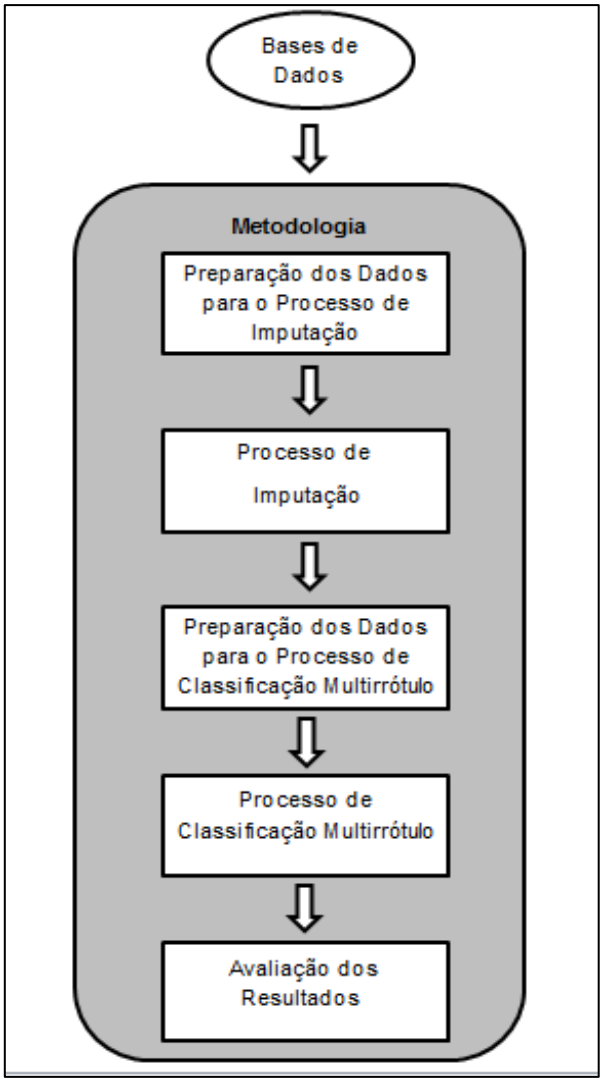
Os valores das instâncias de um arquivo ARFF devem respeitar os tipos de dados dos atributos a que correspondem. Como na Figura 16, os valores das classes geralmente são 0 ou 1, sendo 0 quando o exemplo não pertence a determinada classe e 1 caso contrário.

3.2 METODOLOGIA

A metodologia para a realização dos experimentos segue cinco etapas básicas: preparação dos dados para o processo de imputação, processo de imputação, preparação dos dados para o processo de classificação, processo de

classificação multirrótulo e avaliação dos resultados, conforme é mostrado na Figura 17.

Figura 17 - Metodologia para a realização dos experimentos



Fonte: Autoria Própria

Cada etapa da metodologia utilizada para a realização dos experimentos é explicada detalhadamente ainda neste capítulo. Em todas as etapas, com exceção do processo de classificação multirrótulo e avaliação dos resultados, foram desenvolvidos algoritmos específicos para atender as necessidades de cada uma. Para a implementação dos algoritmos foi utilizada a linguagem C e para a compilação foi usado o compilador GCC versão 5.4.0 que é um software livre disponibilizado pela GNU Compiler Collection (https://gcc.gnu.org/). O sistema operacional utilizado foi o Ubuntu 16.04 LTS. Os experimentos foram executados em

uma máquina com processador Intel Core i5-4210U com clock de 1,70 GHz e 8 GB de memória RAM.

3.2.1 Preparação dos Dados para o Processo de Imputação

Antes de ocorrer o processo de imputação nas bases de dados selecionadas deve acontecer a preparação dos dados. É importante que todas as bases de dados estejam no mesmo formato. Neste trabalho, os arquivos referentes a bases de dados devem estar no formato ARFF. O objetivo da padronização das bases de dados é facilitar o processo de imputação de valores, no qual são implementados algoritmos de imputação para um único padrão, problema que será tratado na Seção 3.2.1.1.

A imputação de dados pode ser aplicada somente em bases de dados com valores faltantes, porém as bases selecionadas para os experimentos estão completas, ou seja, com todos os valores preenchidos. Portanto, as bases de dados completas precisam ser transformadas em bases de dados incompletas, ou seja, com valores ausentes para que, em seguida, ocorra a imputação dos dados, conforme é mostrado na Seção 3.2.1.2.

3.2.1.1 Padronização das bases de dados

Após a seleção das seis bases de dados multirrótulo constatou-se que nem todas estavam no formato ARFF apresentado na Figura 16. Das seis bases de dados selecionadas, três estavam no formato ARFF para MULAN, sendo esta última uma biblioteca JAVA para aprendizado multirrótulo.

No formato ARFF para MULAN deve ser colocado na frente do número de classes um hífen (-) para especificar que o arquivo ARFF é para MULAN. A lista de atributos apresenta as classes por último, ao invés de primeiro como ocorre no formato ARFF da Figura 16. A Figura 18 mostra um exemplo de arquivo ARFF para MULAN referente a uma base de dados multirrótulo

Figura 18 - Exemplo de arquivo no formato ARFF para MULAN

```
1  @relation 'Nome_Base: -C -3'
2
3  @attribute x1 numeric
4  @attribute x2 numeric
5  @attribute x3 numeric
6  @attribute x4 numeric
7  @attribute classe1 {0,1}
8  @attribute classe2 {0,1}
9  @attribute classe3 {0,1}
10
11 @data
12 0.13,0.07,0.22,0.60,0,1,1
13 0.54,0.35,0.15,0.79,0,1,0
14 0.17,0.24,0.25,0.43,0,0,1
```

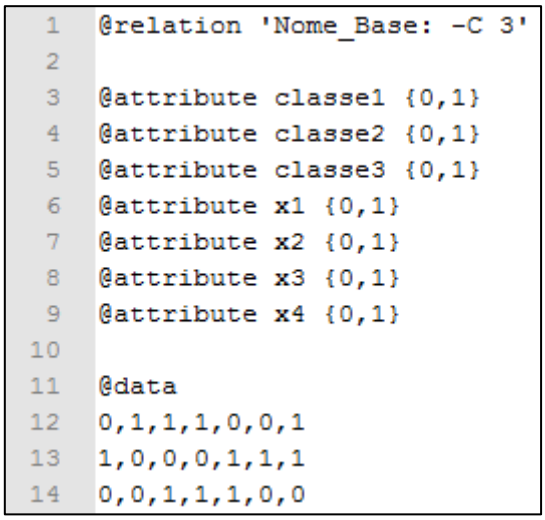
Fonte: Autoria Própria

Na Figura 18, a linha prefixada com *@relation* se refere as características da base de dados, onde Nome_Base é o nome do conjunto e 3 é a sua quantidade de classes. O número de classes é definido após -C. Há um hífen (-) na frente do número de classes especificando que o arquivo ARFF é para MULAN. Na lista de atributos (linhas especificadas com *@attribute*), inicialmente são apresentados os atributos e depois as classes. Após a demarcação *@data* tem-se a lista de instâncias da base de dados. Neste caso, os quatro primeiros valores de uma instância são referentes aos atributos x1, x2, x3 e x4 e os três últimos valores às classes classe1, classe2 e classe3 respectivamente.

As bases de dados *Mediamill*, *Medical e Yeast* estavam no formato ARFF para MULAN e foram transformadas para o formato ARFF comum, deixando todas as bases de dados em um único padrão.

Outro problema encontrado é que algumas bases de dados estavam no formato ARFF, mas em versão resumida. Bases resumidas ocorrem em bases de dados binárias, ou seja, em bases onde os atributos e classes podem receber apenas os valores zero e um. A Figura 19 apresenta um exemplo de base de dados binária.

Figura 19 - Exemplo de base de dados binária



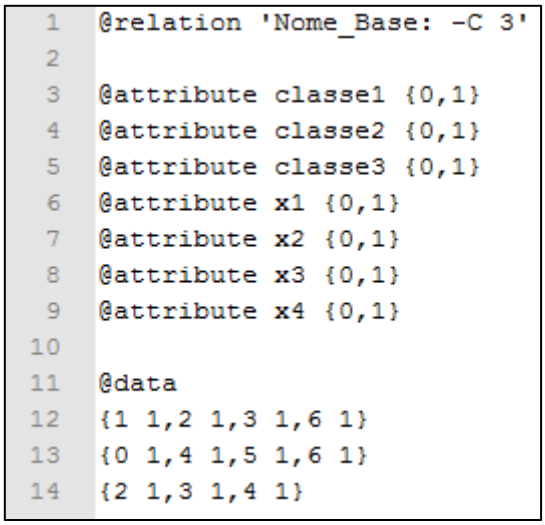
```
1  @relation 'Nome_Base: -C 3'
2
3  @attribute classe1 {0,1}
4  @attribute classe2 {0,1}
5  @attribute classe3 {0,1}
6  @attribute x1 {0,1}
7  @attribute x2 {0,1}
8  @attribute x3 {0,1}
9  @attribute x4 {0,1}
10
11 @data
12 0,1,1,1,0,0,1
13 1,0,0,0,1,1,1
14 0,0,1,1,1,0,0
```

Fonte: Autoria Própria

Na Figura 19 é possível observar que as classes e atributos do conjunto podem receber somente os valores zero e um {0,1}, o que caracteriza esta base de dados como binária. Portanto, todos os valores das instâncias são zero e um, respeitando aos valores permitidos para os atributos e classes correspondentes.

A diferença entre bases de dados comuns e resumidas se encontra nas instâncias. Em instâncias de bases de dados resumidas não são apresentados *n* valores referentes aos *n* atributos existentes na base, como ocorre em bases comuns. A Figura 20 mostra a versão resumida da base de dados binária apresentada na Figura 19.

Figura 20 - Exemplo de base de dados binária em versão resumida



```
1  @relation 'Nome_Base: -C 3'
2
3  @attribute classe1 {0,1}
4  @attribute classe2 {0,1}
5  @attribute classe3 {0,1}
6  @attribute x1 {0,1}
7  @attribute x2 {0,1}
8  @attribute x3 {0,1}
9  @attribute x4 {0,1}
10
11 @data
12 {1 1,2 1,3 1,6 1}
13 {0 1,4 1,5 1,6 1}
14 {2 1,3 1,4 1}
```

Fonte: Autoria Própria

Como pode-se observar na Figura 20, uma instância de uma base de dados resumida informa apenas os números referentes aos atributos que possuem valor igual a um. Neste caso, classe1 corresponde ao atributo 0, classe2 é o atributo 1 e assim sucessivamente até que o atributo x4 equivale ao atributo 6. Na primeira instância deste caso, os valores dos atributos 1, 2, 3 e 6 equivalem a um, consequentemente os demais atributos valem zero. Em outras palavras, os valores de classe1, x2 e x3 são zero e classe2, classe3, x1 e x4 valem um.

As bases de dados *Enron* e *Medical* estavam em versão resumida e foram transformadas em bases de dados comuns, ou seja, bases que possuem instâncias com *n* valores para *n* atributos existentes no conjunto de dados.

3.2.1.2 Transformação de bases de dados completas em incompletas

Foram selecionadas seis bases de dados que estão completas, ou seja, sem valores ausentes. Como o objetivo deste trabalho é avaliar técnicas de imputação aplicadas em bases de dados multirrótulo, é necessário que haja valores faltantes nas bases para que técnicas de imputação possam ser executadas.

Em cada instância das bases de dados foram retirados 10%, 20% e 30% dos valores aleatoriamente, deixando as bases de dados com valores faltantes. A Tabela 2 mostra a quantidade de atributos retirados em cada instância das bases de dados completas para torná-las incompletas.

Tabela 2 - Quantidade de atributos retirados por instância

Bases de Dados	Quant. de Atributos em cada Instância	-10%	-20%	-30%
Emotions	71	7	14	21
Enron	1.001	100	200	300
Mediamill	120	12	24	36
Medical	1.449	144	289	434
Scene	294	29	58	88
Yeast	103	10	20	30

Fonte: Autoria própria

A Tabela 2 apresenta a quantidade de atributos em cada instância das bases de dados e as colunas -10%, -20% e -30% se referem às quantidades de

atributos retirados de cada instância de acordo com as porcentagens. Considere a base de dados *Medical* que possui a maior quantidade de atributos em cada instância, ou seja, 1.449. Para transformar 10% dos dados desta base em valores ausentes, cada instância deve ter 144 atributos removidos. Para retirar 20% dos dados, cada instância da base precisa ter 289 atributos excluídos. Por fim, para tornar 30% dos dados da *Medical* em valores faltantes, em cada instância 434 atributos devem ser retirados.

A Tabela 2 apresenta as quantidades de atributos removidos em cada instância de uma base de dados. Para saber o número total de atributos removidos em uma determinada base de dados, é necessário multiplicar o número de atributos removidos em cada instância pelo número de instâncias existentes na base, já que uma base de dados não é formada apenas por uma instância, mas sim por um conjunto de instâncias. As quantidades totais de atributos removidos nas bases de dados de acordo com as porcentagens 10%, 20% e 30% são mostradas na Tabela 3.

Tabela 3 - Quantidade de atributos retirados por base de dados

Base de Dados	Quant. de Atributos em cada Instância	Total de Instâncias	Total de Atributos	-10%	-20%	-30%
Emotions	71	592	42.032	4.144	8.288	12.432
Enron	1.001	1.702	1.703.702	170.200	340.400	510.600
Mediamill	120	43.907	5.268.840	526.884	1.053.768	1.580.652
Medical	1.449	978	1.417.122	140.832	282.642	424.452
Scene	294	2.407	707.658	69.803	139.606	211.816
Yeast	103	2.417	248.951	24.170	48.340	72.510

Fonte: Autoria própria

Na Tabela 3 são apresentadas as quantidades de atributos presentes em cada instância (2ª coluna) das bases de dados (1ª coluna), quantidades de instâncias existentes nas bases de dados (3ª coluna) e quantidade total de atributos nas base de dados (4ª coluna), sendo esta última calculada a partir da multiplicação das duas colunas anteriores. As quantidades totais de atributos retirados de uma base de dados de acordo com as porcentagens escolhidas estão representadas pelas colunas -10%, -20% e -30% e são obtidas através da multiplicação das

quantidades de atributos retirados em cada instância da base (conforme apresentado na Tabela 2) pela quantidade total de instâncias da mesma base.

3.2.2 Processo de Imputação

Após ser feita a preparação dos dados, deve-se utilizar diferentes algoritmos de imputação nas bases de dados com valores faltantes para que os resultados gerados possam ser comparados.

As bases de dados com 10%, 20% e 30% de seus valores faltantes passaram pelos processos de Imputação pela Moda, Imputação pela Média, Imputação pela Mediana e Imputação KNN Iterativo. Com isso, as bases de dados ficaram completas.

Os algoritmos de imputação foram executados uma vez para cada base de dados que tinha valores faltantes. O KNN Iterativo foi aplicado em um único ciclo com $k = 3$, ou seja, para cada instância com valores ausentes foram encontrados três vizinhos mais próximos.

3.2.3 Preparação dos Dados para o Processo de Classificação Multirrótulo

É necessário validar a eficácia dos algoritmos de imputação, para isso devem ser aplicados os métodos de classificação multirrótulo sob as bases de dados imputadas. Para que ocorra o processo de classificação, as bases de dados precisam ser divididas em teste e treinamento.

Como o objetivo deste trabalho é fazer uma análise da aplicação de algoritmos de imputação em bases de dados multirrótulo, todos os algoritmos de classificação devem ser executados sob as mesmas condições, ou seja, utilizando os mesmos conjuntos de treinamento e teste para uma mesma base com diferentes porcentagens de valores faltantes.

Para dividir as bases de dados em treinamento e teste foi utilizado o método *Holdout*. Nesse método 2/3 das instâncias da base é destinado para treinamento e 1/3 para teste. O conjunto de treinamento é usado para construir o classificador e o conjunto de teste serve para validar o classificador (OLSON; DELEN, 2008). A

Tabela 4 mostra como ficou a divisão das bases de dados em conjuntos de treinamento e teste utilizando o método *Holdout*.

Tabela 4 - Divisão das bases em treinamento e teste

Base	Instâncias	Treinamento (2/3)	Teste (1/3)
Emotions	592	395	197
Enron	1.702	1.135	567
Mediamill	43.907	29.272	14.635
Medical	978	652	326
Scene	2.407	1.605	802
Yeast	2.417	1.612	805

Fonte: Autoria própria

Conforme é demonstrado na Tabela 4, para a base de dados *Emotions* que contém ao todo 592 instâncias, o conjunto de treinamento ficou com 395 instâncias que equivalem a 2/3 do número de instâncias e para teste foram separadas 197 instâncias equivalentes a 1/3 do número de instâncias. A divisão dos demais conjuntos segue a mesma lógica da base *Emotions*.

3.2.4 Processo de Classificação Multirrótulo

Os algoritmos de classificação multirrótulo selecionados para validar os algoritmos de imputação são: BR, LP e RAKEL. Como estes algoritmos transformam um problema multirrótulo em vários problemas monorrótulo, devem ser escolhidos algoritmos de classificação monorrótulo para serem combinados com os classificadores multirrótulo. Os algoritmos de classificação monorrótulo que foram escolhidos são o *Naive Bayes* e o J48.

Para a realização do processo de classificação e avaliação dos resultados utilizou-se o *framework* Meka que possui implementados os algoritmos de classificação selecionados para a realização dos experimentos e também as medidas de avaliação de classificadores multirrótulo.

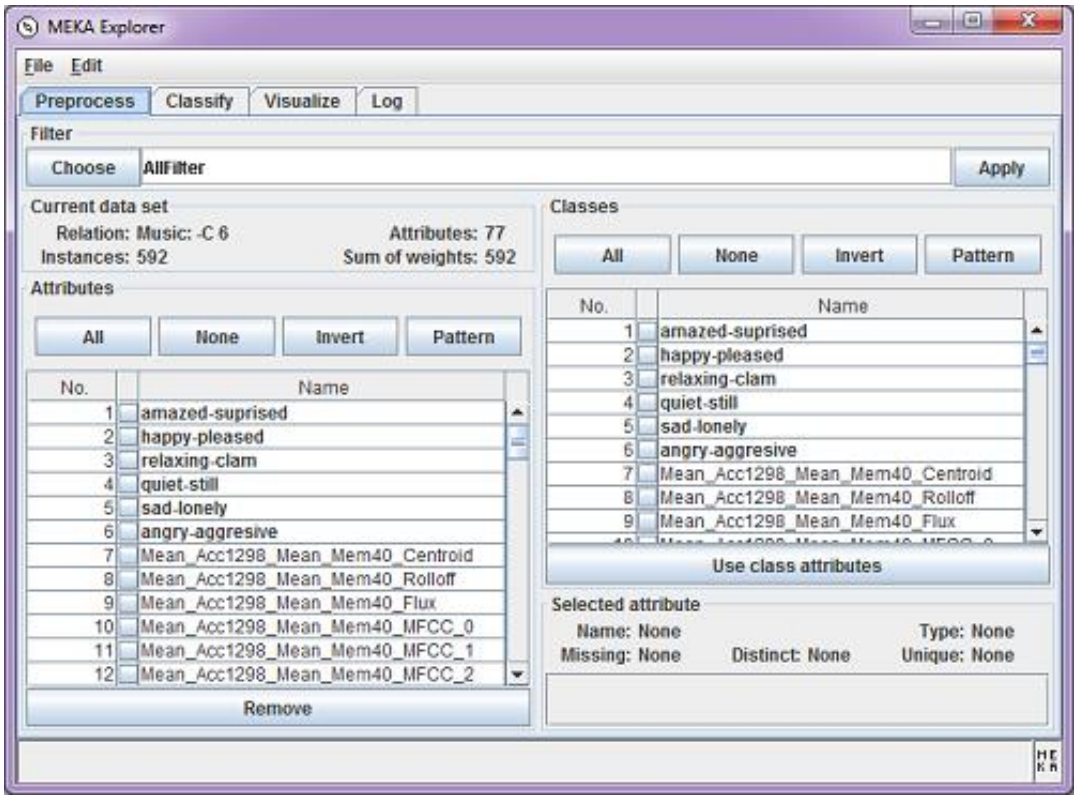
A classificação deve ocorrer utilizando as bases de dados imputadas e também as bases de dados originais a fim de comparar se os resultados obtidos pelos algoritmos de imputação se aproximaram dos resultados das bases de dados originais.

3.2.4.1 *Framework* Meka

O *framework* Meka é uma extensão do *framework* Weka. Ambos são utilizados para o processo de classificação de dados. A diferença é que o Weka é destinado para problemas monorrótulo, enquanto que o Meka resolve problemas multirrótulo.

O Meka é um projeto *open source* de implementação de vários métodos de classificação e avaliação de problemas multirrótulo que permite duas formas de uso: linha de comando ou interface gráfica (GUI). O *framework* pode ser executado em máquinas com Java instalado (versão 1.7 ou acima) utilizando os sistemas operacionais Linux, OSX e Windows. (READ *et al*, 2016).

Figura 21 - Interface Meka: carregamento e configuração de bases de dados

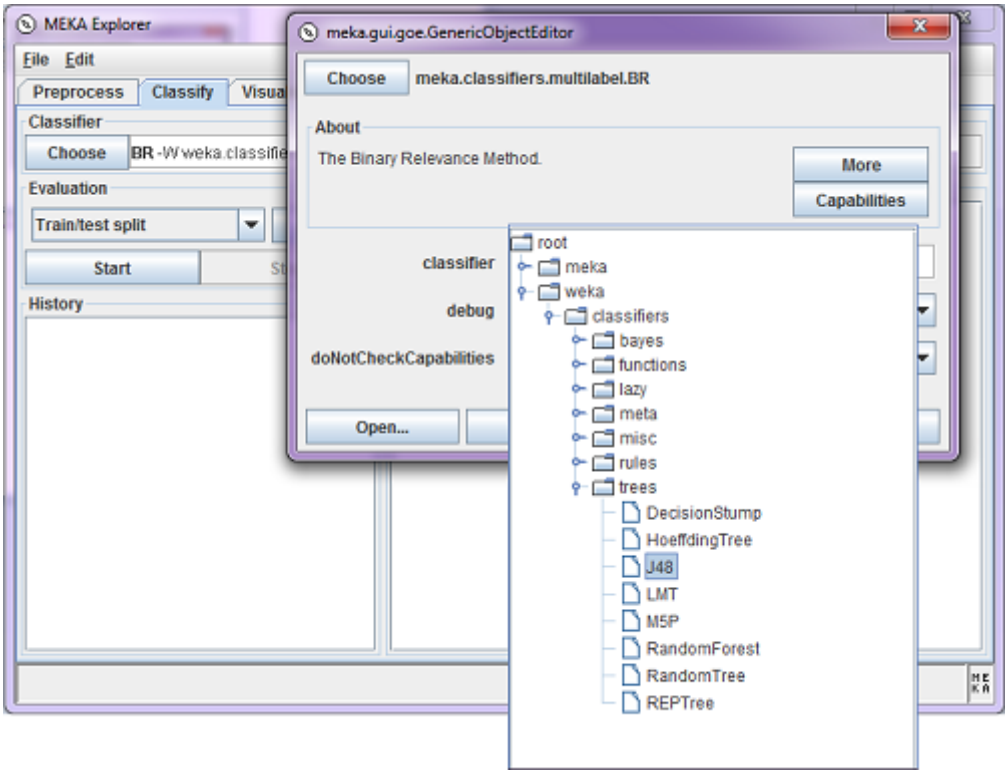


Fonte: Autoria Própria

A Figura 21 mostra a GUI do Meka para o carregamento e a configuração de um conjunto de dados. Para este exemplo foi utilizada a base de dados *Emotions*. Após o carregamento dos dados na memória, são setadas as quantidades de classes, instâncias e atributos da base de dados. É possível observar que a base de

dados *Emotions* possui 6 classes, 592 instâncias e 77 atributos (incluindo o número de classes).

Figura 22 - Interface Meka: classificação de dados



Fonte: Autoria Própria

As GUIs do Meka para o processo de classificação de dados podem ser visualizadas na Figura 22. Através destas interfaces é possível escolher um método multirrótulo e um método monorrótulo. É necessário escolher um método monorrótulo quando o método multirrótulo escolhido for de transformação de um problema multirrótulo para vários problemas monorrótulo, como por exemplo, os métodos BR, LP e RAKEL.

3.2.5 Avaliação dos Resultados

Foram utilizadas as medidas Acurácia, *Hamming Loss* (HL) e Medida F obtidas através do *framework* Meka para avaliar os classificadores. Ainda não há um consenso sobre qual medida de avaliação é mais eficiente (SILVA, 2013). Portanto, serão realizadas análises considerando os resultados que tiveram maiores vantagens em relação a estas medidas.

4 RESULTADOS EXPERIMENTAIS

Os resultados obtidos com os experimentos estão separados em três seções. Primeiramente, são mostrados os resultados referentes às bases de dados completas e originais. Em seguida, são apresentados os resultados relativos às bases de dados com valores imputados e suas comparações em relação aos resultados das bases de dados originais. Por fim, é feita uma comparação entre todos os resultados obtidos com bases de dados imputadas com o objetivo de mostrar o melhor algoritmo de imputação no geral.

4.1 RESULTADOS BASES DE DADOS COMPLETAS

As bases de dados completas e originais passaram pelo processo de classificação multirrótulo para que os valores das medidas de avaliação alcançados pudessem ser comparados com os valores obtidos por classificadores que utilizaram bases de dados imputadas. Com isso, é possível averiguar quais algoritmos de imputação permitiram alcançar resultados mais próximos aos das bases de dados originais.

Todas as bases de dados originais foram submetidas aos classificadores multirrótulo BR, LP e RAKEL. Para cada classificador multirrótulo foram executados os classificadores monorrótulo J48 e *Naive Bayes* (NB). As medidas Acurácia, HL e Medida F avaliaram a eficácia dos classificadores escolhidos.

O Quadro 8 apresenta os resultados das medidas de avaliação Acurácia, HL e Medida F para os classificadores aplicados na base de dados *Emotions*.

Quadro 8 - Avaliação *Emotions*

Base de Dados *Emotions*						
Medida	**BR**		**LP**		**RAKEL**	
Avaliação	**J48**	**NB**	**J48**	**NB**	**J48**	**NB**
Acurácia	0,459	0,511	0,466	0,514	0,514	0,545
HL	0,281	0,209	0,261	0,229	0,251	0,225
Medida F	0,618	0,663	0,578	0,623	0,643	0,666

Fonte: Autoria Própria

O Quadro 9 apresenta os resultados das medidas de avaliação Acurácia, HL e Medida F para os classificadores aplicados na base de dados *Enron*.

Quadro 9 - Avaliação *Enron*

Base de Dados *Enron*						
Medida Avaliação	**BR**		**LP**		**RAKEL**	
	J48	**NB**	**J48**	**NB**	**J48**	**NB**
Acurácia	0,402	0,186	0,333	0,415	0,029	0,049
HL	0,058	0,088	0,073	0,059	0,064	0,105
Medida F	0,552	0,309	0,419	0,514	0,061	0,085

Fonte: Autoria Própria

O Quadro 10 apresenta os resultados das medidas de avaliação Acurácia, HL e Medida F para os classificadores aplicados na base de dados *Mediamill*.

Quadro 10 - Avaliação *Mediamill*

Base de Dados *Mediamill*						
Medida Avaliação	**BR**		**LP**		**RAKEL**	
	J48	**NB**	**J48**	**NB**	**J48**	**NB**
Acurácia	0,072	0,193	0,378	0,310	0,129	0,067
HL	0,313	0,105	0,043	0,054	0,044	0,105
Medida F	0,135	0,289	0,488	0,454	0,179	0,124

Fonte: Autoria Própria

O Quadro 11 apresenta os resultados das medidas de avaliação Acurácia, HL e Medida F para os classificadores aplicados na base de dados *Medical*.

Quadro 11 - Avaliação *Medical*

Base de Dados *Medical*						
Medida Avaliação	**BR**		**LP**		**RAKEL**	
	J48	**NB**	**J48**	**NB**	**J48**	**NB**
Acurácia	0,757	0,362	0,733	0,396	0,466	0,242
HL	0,011	0,029	0,014	0,029	0,018	0,025
Medida F	0,809	0,470	0,753	0,420	0,584	0,366

Fonte: Autoria Própria

O Quadro 12 apresenta os resultados das medidas de avaliação Acurácia, HL e Medida F para os classificadores aplicados na base de dados *Scene*.

Quadro 12 - Avaliação *Scene*

Base de Dados *Scene*						
Medida	**BR**		**LP**		**RAKEL**	
	J48	**NB**	**J48**	**NB**	**J48**	**NB**
Acurácia	0,510	0,470	0,575	0,623	0,601	0,603
HL	0,151	0,178	0,149	0,135	0,146	0,145
Medida F	0,596	0,589	0,587	0,639	0,648	0,654

Fonte: Autoria Própria

O Quadro 13 apresenta os resultados das medidas de avaliação Acurácia, HL e Medida F para os classificadores aplicados na base de dados *Yeast*.

Quadro 13 - Avaliação *Yeast*

Base de Dados *Yeast*						
Medida	**BR**		**LP**		**RAKEL**	
	J48	**NB**	**J48**	**NB**	**J48**	**NB**
Acurácia	0,295	0,450	0,392	0,456	0,424	0,423
HL	0,495	0,253	0,283	0,246	0,308	0,291
Medida F	0,452	0,581	0,528	0,583	0,567	0,559

Fonte: Autoria Própria

Os valores das medidas de avaliação para os classificadores aplicados em cada base (Quadros 8, 9, 10, 11, 12 e 13) não serão comparados, porque o objetivo do presente trabalho não é comparar algoritmos de classificação, mas sim de imputação. Serão comparados os resultados das medidas de avaliação das bases de dados originais em relação aos resultados das bases de dados que sofreram imputação e que posteriormente foram utilizadas pelo mesmo classificador.

4.2 RESULTADOS BASES DE DADOS IMPUTADAS

As bases de dados com 10%, 20% e 30% dos seus valores ausentes passaram pelos processos de Imputação pela Moda, Média, Mediana e KNN Iterativo. Em seguida, as bases de dados imputadas foram submetidas aos classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e *Naive Bayes* (NB). As medidas Acurácia, HL e Medida F avaliaram a eficácia dos classificadores.

Para cada base de dados imputada com uma determinada porcentagem de valores foi feita uma comparação entre os seus resultados em relação aos resultados das bases de dados originais para descobrir qual algoritmo de imputação foi melhor, ou seja, se aproximou mais dos valores reais. No fim de cada subseção, foi encontrado o melhor algoritmo de imputação para a base em questão através do cálculo da média das porcentagens dos melhores algoritmos de imputação das suas bases de dados com 10%, 20% e 30% de valores imputados.

Os resultados das medidas de avaliação para os classificadores aplicados nas bases de dados imputadas juntamente com suas comparações serão mostrados de maneira separada para cada base com o objetivo de obter maior clareza na apresentação dos resultados.

4.2.1 Base de Dados *Emotions*

Foram realizados experimentos com três porcentagens diferentes de valores imputados na base de dados *Emotions*, por isso os resultados serão apresentados separadamente para cada porcentagem.

Os passos necessários para fazer a análise dos resultados da base *Emotions* 10% Imputada servirão como modelo para a obtenção dos resultados das demais bases de dados imputadas. Inclusive, quadros e gráficos serão utilizados para as outras bases com os mesmos objetivos e significados, porém com os valores resultantes de cada base de dados.

4.2.1.1 *Emotions* 10% Imputada

O Quadro 14 mostra os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Emotions* com 10% de seus valores imputados.

Quadro 14 - Avaliação *Emotions* 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,418	0,491	0,409	0,483	0,497	0,551
	Média	0,434	0,504	0,436	0,500	0,478	0,529
	Mediana	0,364	0,497	0,438	0,485	0,515	0,534
	KNN	0,370	0,513	0,450	0,493	0,538	0,544
HL	Moda	0,274	0,214	0,287	0,241	0,262	0,219
	Média	0,263	0,209	0,296	0,232	0,266	0,234
	Mediana	0,336	0,212	0,278	0,238	0,251	0,231
	KNN	0,360	0,209	0,276	0,244	0,237	0,228
Medida F	Moda	0,566	0,655	0,540	0,599	0,629	0,669
	Média	0,577	0,663	0,53	0,614	0,606	0,654
	Mediana	0,514	0,658	0,556	0,605	0,640	0,656
	KNN	0,511	0,663	0,555	0,601	0,670	0,663

Fonte: Autoria Própria

No Quadro 14, a coluna Algoritmos de Imputação representa os métodos de Imputação pela Moda, Média, Mediana e KNN Iterativo que foram utilizados para completar a base de dados *Emotions*. Cada base completada com um algoritmo de imputação passou pelos classificadores multirrótulo BR, LP e RAKEL juntamente com os classificadores monorrótulo J48 e NB que estão representados pelas demais colunas do Quadro 14. A partir do processo de classificação obtiveram-se os resultados de Acurácia, HL e Medida F para cada base de dados *Emotions* preenchida por um determinado algoritmo de imputação que passou por um dado classificador.

Para saber quais algoritmos de imputação alcançaram os resultados das medidas de avaliação mais próximos aos da base de dados *Emotions* Original pode-se calcular a diferença das medidas de avaliação da base *Emotions* completa (Quadro 8) em relação a base *Emotions* que teve 10% dos valores imputados (Quadro 14), conforme é mostrado no Quadro 15. Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 15 - Diferença de avaliação *Emotions* Original e 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,041	0,020	0,057	0,031	0,017	-0,006
	Média	0,025	0,007	0,030	0,014	0,036	0,016
	Mediana	0,095	0,014	0,028	0,029	-0,001	0,011
	KNN	0,089	-0,002	0,016	0,021	-0,024	0,001
HL	Moda	0,007	-0,005	-0,026	-0,012	-0,011	0,006
	Média	0,018	0,000	-0,035	-0,003	-0,015	-0,009
	Mediana	-0,055	-0,003	-0,017	-0,009	0,000	-0,006
	KNN	-0,079	0,000	-0,015	-0,015	0,014	-0,003
Medida F	Moda	0,052	0,008	0,038	0,024	0,014	-0,003
	Média	0,041	0,000	0,048	0,009	0,037	0,012
	Mediana	0,104	0,005	0,022	0,018	0,003	0,010
	KNN	0,107	0,000	0,023	0,022	-0,027	0,003

Fonte: Autoria Própria

No Quadro 15, quanto mais próximo de zero for um valor, mais próximo do valor real ele está. Foram analisados os resultados de cada classificador, ou seja, foi verificado se o valor mais próximo de zero das medidas de avaliação de um classificador foi alcançado por bases de dados que utilizaram Imputação pela Moda, Média, Mediana ou KNN Iterativo. Por exemplo, considerando o classificador BR com J48, a Acurácia que está mais próxima de zero foi alcançada pela base *Emotions* que sofreu Imputação pela Média. Já a medida HL mais próxima do valor real foi obtida através da base *Emotions* que foi imputada pela Moda. Por fim, a Medida F que mais se aproximou do valor real resultou da base de dados *Emotions* que utilizou a Imputação pela Média.

Para descobrir qual algoritmo de imputação foi melhor para um classificador aplicado em uma base de dados, ou seja, teve resultados mais próximos aos de uma base de dados original deve-se verificar a imputação que obteve a maior quantidade de resultados das medidas de avaliação mais próximos de zero para o classificador em questão. Por exemplo, no Quadro 15, o algoritmo de Imputação pela Média obteve dois melhores resultados das medidas de avaliação para o classificador BR com J48, sendo Acurácia e Medida F, enquanto que Imputação pela Moda teve apenas o melhor resultado de HL. Portanto, Imputação pela Média conseguiu a maior quantidade de medidas de avaliação mais próximas dos valores

reais e por isso é considerada a melhor imputação para o classificador BR com j48 aplicado na base *Emotions* 10% Imputada.

O Quadro 16 mostra um resumo para a base *Emotions* 10% Imputada com os melhores algoritmos de imputação para cada classificador, de acordo com o Quadro 15.

Quadro 16 - Melhores algoritmos de imputação para classificadores da *Emotions* 10% Imputada

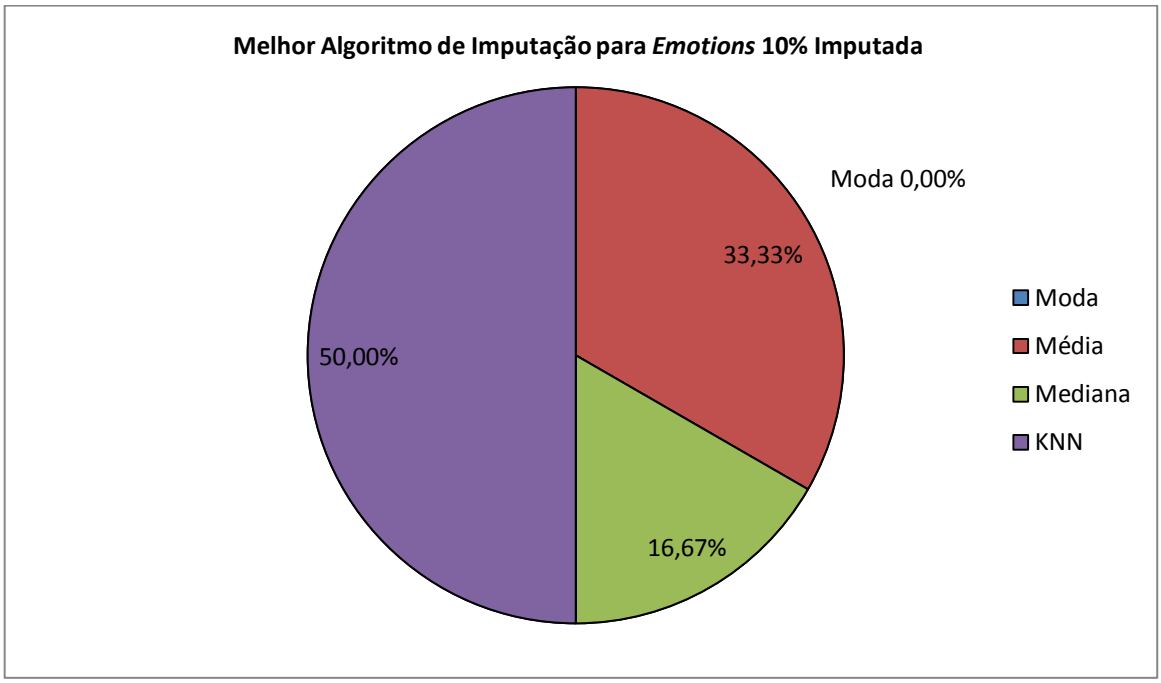
Classificador	Algoritmo de Imputação
BR com J48	Média
BR com NB	KNN
LP com J48	KNN
LP com NB	Média
RAKEL com J48	Mediana
RAKEL com NB	KNN

Fonte: Autoria Própria

No Quadro 16, considere para exemplo o classificador BR com NB. O melhor algoritmo de imputação foi KNN Iterativo, pois obteve 3 melhores resultados das medidas de avaliação, sendo Acurácia, HL e Medida F contra 2 melhores resultados obtidos pelo algoritmo de Imputação pela Média que foram HL e Medida F, conforme pode ser visto no Quadro 15. Para encontrar o melhor algoritmo de imputação para os demais classificadores seguiu-se a mesma lógica. Foi verificado o algoritmo de imputação que obteve o maior número de melhores resultados entre as medidas de avaliação Acurácia, HL e Medida F.

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 16), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Emotions* 10% Imputada, conforme é mostrado no Gráfico 1.

Gráfico 1 - Melhor algoritmo de imputação para *Emotions* 10% Imputada



Fonte: Autoria Própria

O Gráfico 1 mostra que o algoritmo Imputação KNN Iterativo foi o melhor para a base de dados *Emotions* 10% Imputada, pois obteve 3 melhores resultados entre 6 casos possíveis (Quadro 16), representando 50% dos resultados mais próximos aos da base de dados *Emotions* Original. Os algoritmos de Imputação pela Média e Mediana conseguiram porcentagens consideráveis de resultados mais próximos aos reais. O pior algoritmo para a base *Emotions* 10% Imputada foi a Imputação pela Moda, representando 0% do total, pois não foi considerado o melhor algoritmo de imputação para nenhum classificador aplicado na base de dados *Emotions* 10% Imputada, conforme pode ser visualizado no Quadro 16.

4.2.1.2 *Emotions* 20% Imputada

O Quadro 17 apresenta os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Emotions* com 20% de seus valores imputados.

Quadro 17 - Avaliação *Emotions* 20% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,421	0,483	0,443	0,505	0,483	0,540
	Média	0,425	0,489	0,411	0,500	0,515	0,543
	Mediana	0,414	0,487	0,376	0,498	0,481	0,539
	KNN	0,425	0,496	0,467	0,482	0,522	0,548
HL	Moda	0,283	0,224	0,280	0,234	0,291	0,223
	Média	0,312	0,223	0,298	0,234	0,261	0,219
	Mediana	0,321	0,223	0,324	0,234	0,277	0,223
	KNN	0,341	0,216	0,267	0,244	0,258	0,223
Medida F	Moda	0,567	0,638	0,546	0,619	0,606	0,660
	Média	0,566	0,641	0,522	0,616	0,629	0,661
	Mediana	0,559	0,641	0,486	0,615	0,613	0,654
	KNN	0,566	0,652	0,573	0,596	0,654	0,662

Fonte: Autoria Própria

O Quadro 18 mostra a diferença das medidas de avaliação para a base *Emotions* completa (Quadro 8) em relação a base *Emotions* que teve 20% dos valores imputados (Quadro 17). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 18 - Diferença de avaliação *Emotions* Original e 20% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,038	0,028	0,023	0,009	0,031	0,005
	Média	0,034	0,022	0,055	0,014	-0,001	0,002
	Mediana	0,045	0,024	0,090	0,016	0,033	0,006
	KNN	0,034	0,015	-0,001	0,032	-0,008	-0,003
HL	Moda	-0,002	-0,015	-0,019	-0,005	-0,040	0,002
	Média	-0,031	-0,014	-0,037	-0,005	-0,010	0,006
	Mediana	-0,040	-0,014	-0,063	-0,005	-0,026	0,002
	KNN	-0,060	-0,007	-0,006	-0,015	-0,007	0,002
Medida F	Moda	0,051	0,025	0,032	0,004	0,037	0,006
	Média	0,052	0,022	0,056	0,007	0,014	0,005
	Mediana	0,059	0,022	0,092	0,008	0,030	0,012
	KNN	0,052	0,011	0,005	0,027	-0,011	0,004

Fonte: Autoria Própria

O Quadro 19 demonstra quais foram os melhores algoritmos de imputação para cada classificador aplicado na base *Emotions* 20% Imputada, de acordo com o Quadro 18.

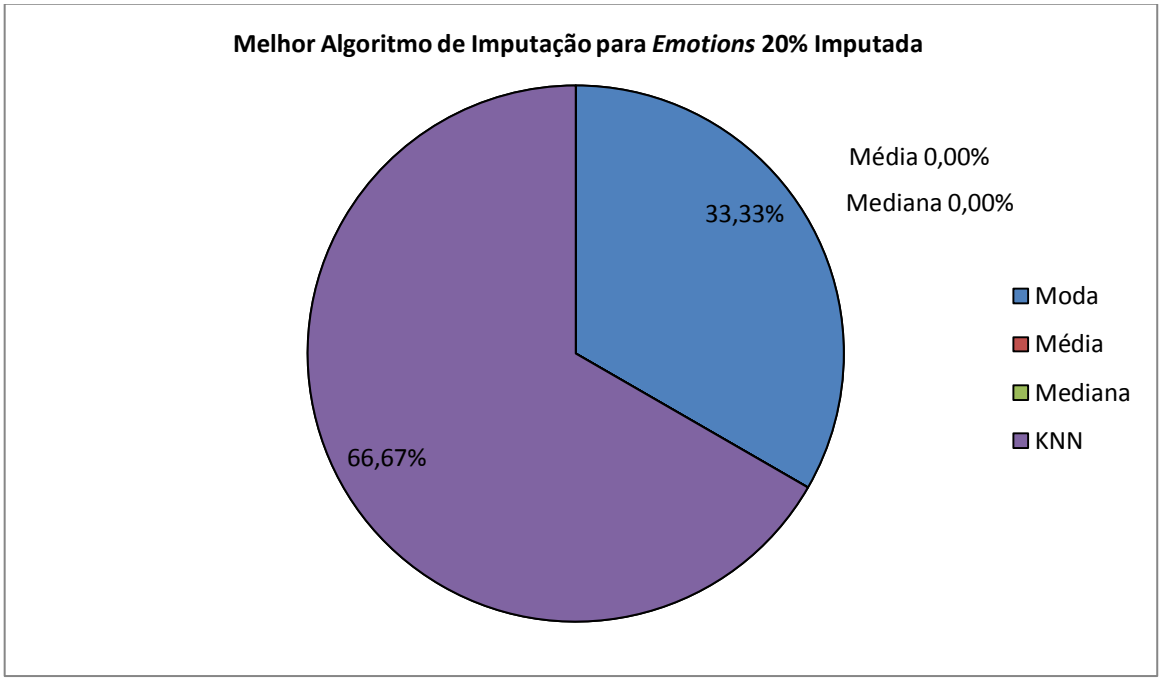
Quadro 19 - Melhores algoritmos de imputação para classificadores da *Emotions* 20% Imputada

Classificador	Algoritmo de Imputação
BR com J48	Moda
BR com NB	KNN
LP com J48	KNN
LP com NB	Moda
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 19), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Emotions* 20% Imputada, conforme é mostrado no Gráfico 2.

Gráfico 2 - Melhor algoritmo de imputação para *Emotions* 20% Imputada



Fonte: Autoria Própria

O Gráfico 2 mostra que a base *Emotions* 20% teve bons resultados com dois algoritmos de Imputação: KNN Iterativo e Moda. Imputação KNN Iterativo se

destacou neste caso, pois obteve 66,67% dos resultados mais próximos aos da base de dados *Emotions* Original. O algoritmo de Imputação pela Moda representa 33,33% dos melhores resultados. Os algoritmos de imputação pela Média e Mediana tiveram os piores resultados, representando 0% do total.

4.2.1.3 *Emotions* 30% Imputada

O Quadro 20 apresenta os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Emotions* com 30% de seus valores imputados.

Quadro 20 - Avaliação *Emotions* 30% Imputada

	Algoritmos de Imputação	**BR**		**LP**		**RAKEL**	
		J48	**NB**	**J48**	**NB**	**J48**	**NB**
Acurácia	**Moda**	0,408	0,474	0,445	0,506	0,490	0,493
	Média	0,440	0,478	0,427	0,480	0,495	0,498
	Mediana	0,436	0,477	0,429	0,483	0,497	0,484
	KNN	0,437	0,486	0,508	0,486	0,513	0,521
HL	**Moda**	0,316	0,233	0,275	0,224	0,284	0,252
	Média	0,300	0,233	0,294	0,240	0,296	0,255
	Mediana	0,304	0,234	0,283	0,239	0,265	0,257
	KNN	0,267	0,223	0,242	0,245	0,253	0,234
Medida F	**Moda**	0,561	0,625	0,547	0,634	0,607	0,620
	Média	0,598	0,625	0,533	0,603	0,620	0,617
	Mediana	0,591	0,622	0,535	0,603	0,628	0,608
	KNN	0,579	0,641	0,622	0,595	0,634	0,641

Fonte: Autoria Própria

O Quadro 21 mostra a diferença das medidas de avaliação para a base *Emotions* completa (Quadro 8) em relação a base *Emotions* que teve 30% dos valores imputados (Quadro 20). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 21 - Diferença de avaliação *Emotions* Original e 30% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,051	0,037	0,021	0,008	0,024	0,052
	Média	0,019	0,033	0,039	0,034	0,019	0,047
	Mediana	0,023	0,034	0,037	0,031	0,017	0,061
	KNN	0,022	0,025	-0,042	0,028	0,001	0,024
HL	Moda	-0,035	-0,024	-0,014	0,005	-0,033	-0,027
	Média	-0,019	-0,024	-0,033	-0,011	-0,045	-0,030
	Mediana	-0,023	-0,025	-0,022	-0,010	-0,014	-0,032
	KNN	0,014	-0,014	0,019	-0,016	-0,002	-0,009
Medida F	Moda	0,057	0,038	0,031	-0,011	0,036	0,046
	Média	0,020	0,038	0,045	0,020	0,023	0,049
	Mediana	0,027	0,041	0,043	0,020	0,015	0,058
	KNN	0,039	0,022	-0,044	0,028	0,009	0,025

Fonte: Autoria Própria

O Quadro 22 se refere a base *Emotions* 30% Imputada e demonstra quais foram os melhores algoritmos de imputação para cada classificador, de acordo com o Quadro 21.

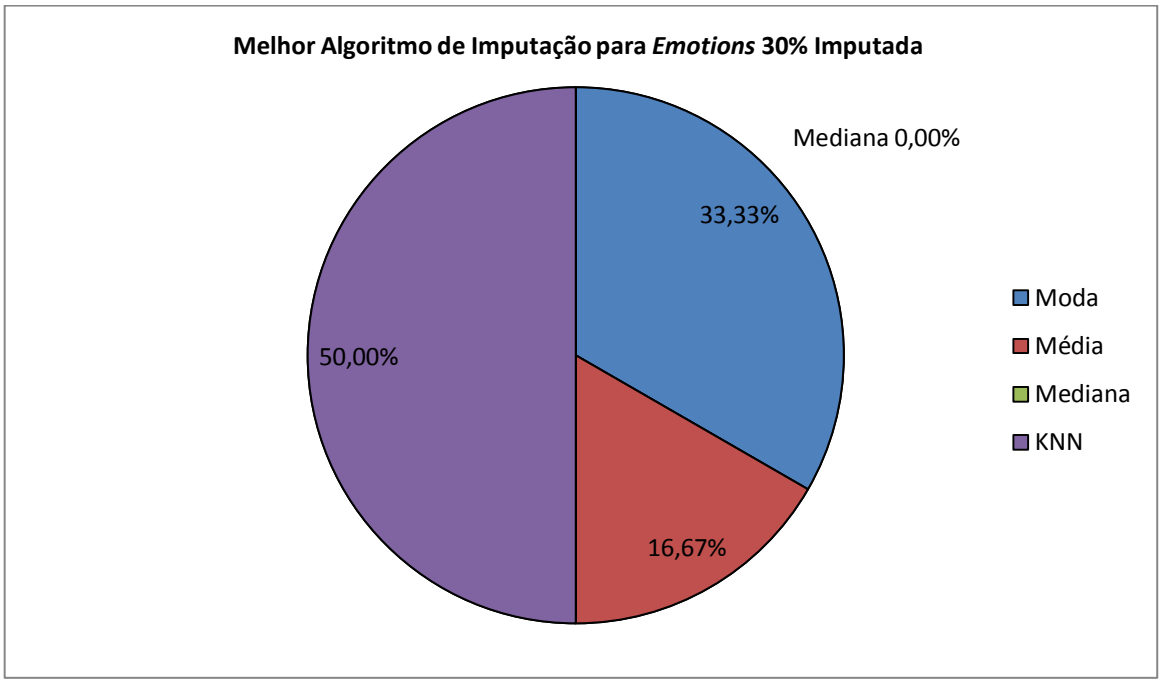
Quadro 22 - Melhores algoritmos de imputação para classificadores da *Emotions* 30% Imputada

Classificador	Algoritmo de Imputação
BR com J48	Média
BR com NB	KNN
LP com J48	Moda
LP com NB	Moda
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 22), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Emotions* 30% Imputada, conforme é mostrado no Gráfico 3.

Gráfico 3 - Melhor algoritmo de imputação para *Emotions* 30% Imputada



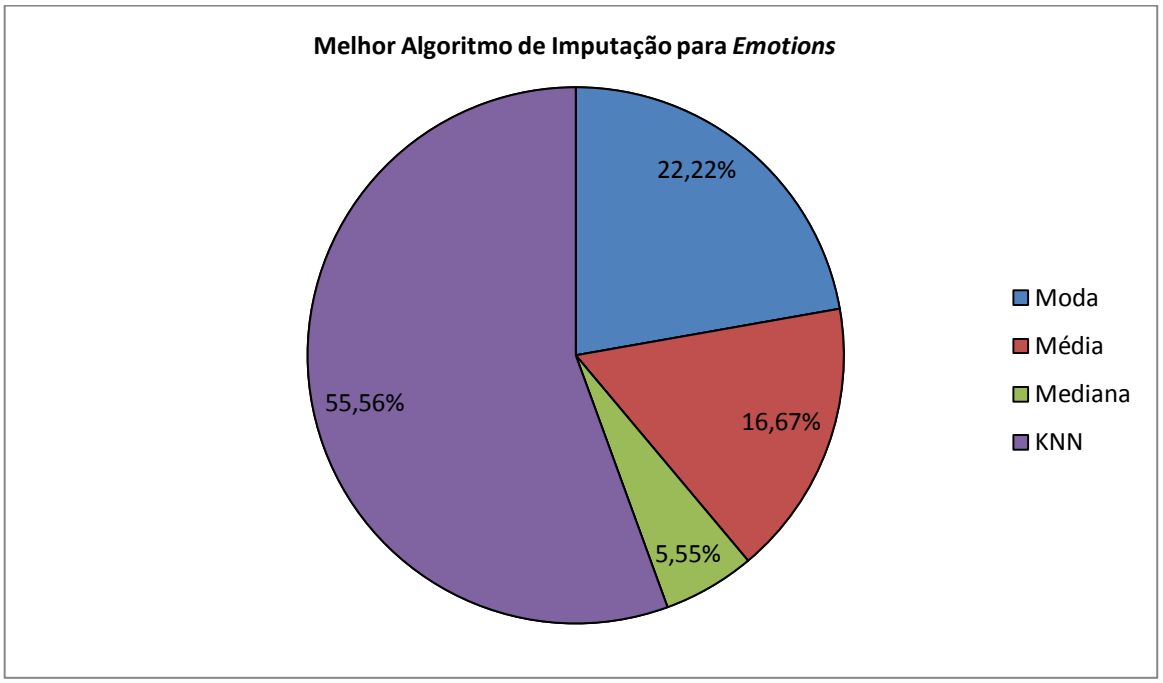
Fonte: Autoria Própria

Pelo Gráfico 3 é possível observar que o algoritmo Imputação KNN Iterativo foi o melhor para a base de dados *Emotions* 30% Imputada, pois obteve 50% dos resultados mais próximos aos da base de dados *Emotions* Original. Os algoritmos de Imputação pela Moda e Média conseguiram resultados significativos, representando juntos 50% dos melhores resultados. O algoritmo de imputação pela Mediana teve o pior resultado, representando 0% de resultados mais próximos aos da base de dados *Emotions* Original.

4.2.1.4 Melhor algoritmo de imputação para *Emotions*

A partir dos Gráficos 1, 2 e 3 pode-se calcular a média das porcentagens dos melhores algoritmos de imputação para as bases de dados *Emotions* com 10%, 20% e 30% de valores imputados e obter o melhor algoritmo de imputação para as bases de dados *Emotions*, conforme é mostrado no Gráfico 4.

Gráfico 4 - Melhor algoritmo de imputação para *Emotions*



Fonte: Autoria Própria

O Gráfico 4 comprova que o algoritmo de Imputação KNN Iterativo alcançou os melhores resultados para as bases de dados *Emotions* que tiveram valores imputados, pois obteve 55,56% dos resultados mais próximos aos da base de dados *Emotions* Original. O segundo melhor algoritmo de imputação para as bases *Emotions* imputadas foi a Moda com 22,22% dos melhores resultados. O terceiro melhor algoritmo de imputação para as bases *Emotions* imputadas foi a Média representando 16,67%. O algoritmo de imputação pela Mediana teve o pior resultado, representando apenas 5,55% dos resultados mais próximos aos reais.

4.2.2 Base de dados *Enron*

Foram realizados experimentos com três porcentagens diferentes de valores imputados na base de dados *Enron*, por isso os resultados serão apresentados separadamente para cada porcentagem.

4.2.2.1 *Enron* 10% Imputada

O Quadro 23 apresenta os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os

classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Enron* com 10% de seus valores imputados.

Quadro 23 - Avaliação *Enron* 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,380	0,201	0,276	0,417	0,020	0,046
	Média	0,383	0,201	0,273	0,416	0,023	0,046
	Mediana	0,380	0,201	0,276	0,417	0,020	0,046
	KNN	0,393	0,194	0,305	0,409	0,027	0,048
HL	Moda	0,061	0,087	0,078	0,058	0,065	0,100
	Média	0,061	0,087	0,078	0,059	0,066	0,100
	Mediana	0,061	0,087	0,078	0,058	0,065	0,100
	KNN	0,058	0,087	0,073	0,060	0,066	0,105
Medida F	Moda	0,521	0,322	0,365	0,514	0,044	0,083
	Média	0,524	0,322	0,361	0,513	0,050	0,083
	Mediana	0,521	0,322	0,365	0,514	0,044	0,083
	KNN	0,543	0,316	0,402	0,510	0,058	0,084

Fonte: Autoria Própria

O Quadro 24 mostra a diferença das medidas de avaliação para a base *Enron* completa (Quadro 9) em relação a base *Enron* 10% Imputada (Quadro 23).

Quadro 24 - Diferença de avaliação *Enron* Original e 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,022	-0,015	0,057	-0,002	0,009	0,003
	Média	0,019	-0,015	0,060	-0,001	0,006	0,003
	Mediana	0,022	-0,015	0,057	-0,002	0,009	0,003
	KNN	0,009	-0,008	0,028	0,006	0,002	0,001
HL	Moda	-0,003	0,001	-0,005	0,001	-0,001	0,005
	Média	-0,003	0,001	-0,005	0,000	-0,002	0,005
	Mediana	-0,003	0,001	-0,005	0,001	-0,001	0,005
	KNN	0,000	0,001	0,000	-0,001	-0,002	0,000
Medida F	Moda	0,031	-0,013	0,054	0,000	0,017	0,002
	Média	0,028	-0,013	0,058	0,001	0,011	0,002
	Mediana	0,031	-0,013	0,054	0,000	0,017	0,002
	KNN	0,009	-0,007	0,017	0,004	0,003	0,001

Fonte: Autoria Própria

O Quadro 25 demonstra quais foram os melhores algoritmos de imputação para cada classificador aplicado na base *Enron* 10% Imputada, de acordo com o Quadro 24.

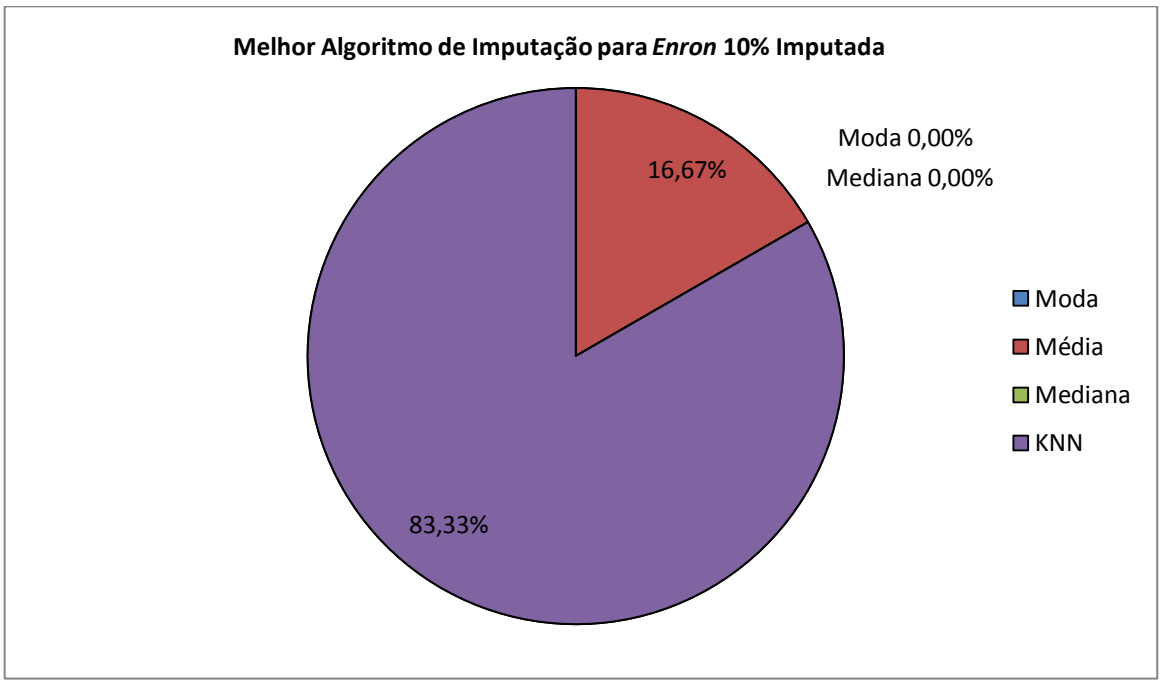
Quadro 25 - Melhores algoritmos de imputação para classificadores da *Enron* 10% Imputada

Classificador	Algoritmo de Imputação
BR com J48	KNN
BR com NB	KNN
LP com J48	KNN
LP com NB	Média
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 25), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Enron* 10% Imputada, conforme é mostrado no Gráfico 5.

Gráfico 5 - Melhor algoritmo de imputação para *Enron* 10% Imputada



Fonte: Autoria Própria

Pelo Gráfico 5 constata-se que o algoritmo Imputação KNN Iterativo atingiu 83,33% dos resultados mais próximos aos da base *Enron* Original e, portanto, foi o

melhor algoritmo de imputação para a *Enron* 10% Imputada. Com 16,67%, o algoritmo de Imputação pela Média representa uma pequena parte dos melhores resultados se comparado ao percentual obtido pelo algoritmo Imputação KNN Iterativo. Os algoritmos de imputação pela Moda e Mediana tiveram os piores resultados, representando 0% dos resultados que mais se aproximaram da base de dados original.

4.2.2.2 *Enron* 20% Imputada

O Quadro 26 apresenta os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Enron* com 20% de seus valores imputados.

Quadro 26 - Avaliação *Enron* 20% Imputada

	Algoritmos de Imputação	**BR**		**LP**		**RAKEL**	
		J48	**NB**	**J48**	**NB**	**J48**	**NB**
Acurácia	**Moda**	0,336	0,217	0,316	0,410	0,019	0,046
	Média	0,336	0,218	0,317	0,415	0,021	0,046
	Mediana	0,336	0,217	0,316	0,410	0,019	0,046
	KNN	0,376	0,204	0,339	0,413	0,022	0,045
HL	**Moda**	0,071	0,084	0,073	0,058	0,066	0,092
	Média	0,070	0,084	0,073	0,058	0,065	0,091
	Mediana	0,071	0,084	0,073	0,058	0,066	0,092
	KNN	0,061	0,086	0,071	0,059	0,065	0,104
Medida F	**Moda**	0,492	0,340	0,395	0,505	0,043	0,085
	Média	0,492	0,342	0,395	0,509	0,046	0,085
	Mediana	0,492	0,340	0,395	0,505	0,043	0,085
	KNN	0,524	0,326	0,422	0,509	0,048	0,080

Fonte: Autoria Própria

O Quadro 27 mostra a diferença das medidas de avaliação para a base *Enron* completa (Quadro 9) em relação a base *Enron* que teve 20% dos valores imputados (Quadro 26). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 27 - Diferença de avaliação *Enron* Original e 20% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,066	-0,031	0,017	0,005	0,010	0,003
	Média	0,066	-0,032	0,016	0,000	0,008	0,003
	Mediana	0,066	-0,031	0,017	0,005	0,010	0,003
	KNN	0,026	-0,018	-0,006	0,002	0,007	0,004
HL	Moda	-0,013	0,004	0,000	0,001	-0,002	0,013
	Média	-0,012	0,004	0,000	0,001	-0,001	0,014
	Mediana	-0,013	0,004	0,000	0,001	-0,002	0,013
	KNN	-0,003	0,002	0,002	0,000	-0,001	0,001
Medida F	Moda	0,060	-0,031	0,024	0,009	0,018	0,000
	Média	0,060	-0,033	0,024	0,005	0,015	0,000
	Mediana	0,060	-0,031	0,024	0,009	0,018	0,000
	KNN	0,028	-0,017	-0,003	0,005	0,013	0,005

Fonte: Autoria Própria

O Quadro 28 se refere a base *Enron* 20% Imputada e demonstra quais foram os melhores algoritmos de imputação para cada classificador, de acordo com o Quadro 27.

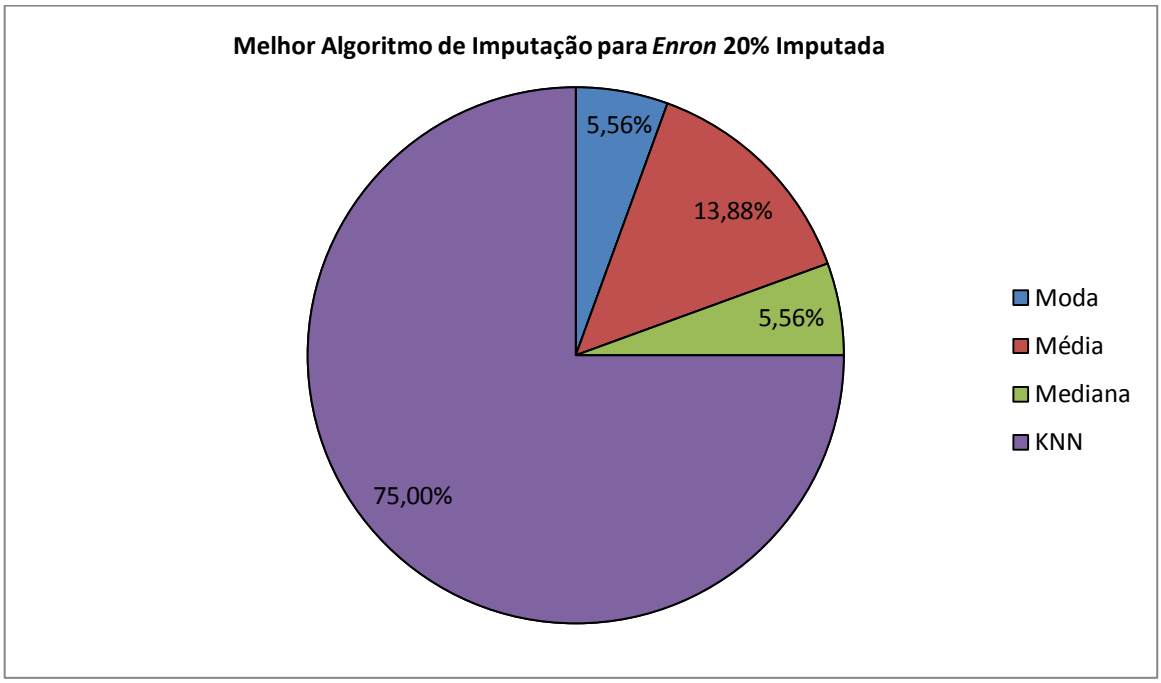
Quadro 28 - Melhores algoritmos de imputação para classificadores da *Enron* 20% Imputada

Classificador	Algoritmo de Imputação
BR com J48	KNN
BR com NB	KNN
LP com J48	KNN
LP com NB	KNN, Média
RAKEL com J48	KNN
RAKEL com NB	Moda, Média, Mediana

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 28), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Enron* 20% Imputada, conforme é mostrado no Gráfico 6.

Gráfico 6 - Melhor algoritmo de imputação para *Enron* 20% Imputada



Fonte: Autoria Própria

O Gráfico 6 evidencia que o algoritmo Imputação KNN Iterativo alcançou a maioria dos melhores resultados, representando 75% dos valores mais próximos aos da base *Enron* Original e, portanto, é considerado o melhor algoritmo de imputação para a base *Enron* 10% Imputada. O algoritmo de Imputação pela Média representa o segundo melhor algoritmo de imputação para a *Enron* 10% Imputada com 13,88% dos melhores resultados. Os algoritmos de imputação pela Moda e Mediana tiveram os piores resultados, representando cada um 5,56% dos resultados que mais se aproximaram da base de dados original.

4.2.2.3 *Enron* 30% Imputada

O Quadro 29 apresenta os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Enron* com 30% de seus valores imputados.

Quadro 29 - Avaliação *Enron* 30% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,313	0,226	0,280	0,412	0,019	0,043
	Média	0,312	0,227	0,277	0,406	0,019	0,043
	Mediana	0,313	0,226	0,280	0,412	0,019	0,043
	KNN	0,327	0,214	0,324	0,401	0,025	0,044
HL	Moda	0,072	0,083	0,078	0,059	0,066	0,084
	Média	0,072	0,083	0,078	0,059	0,066	0,084
	Mediana	0,072	0,083	0,078	0,059	0,066	0,084
	KNN	0,071	0,085	0,073	0,061	0,066	0,101
Medida F	Moda	0,465	0,349	0,367	0,499	0,041	0,084
	Média	0,464	0,350	0,365	0,495	0,041	0,084
	Mediana	0,465	0,349	0,367	0,499	0,041	0,084
	KNN	0,483	0,333	0,413	0,491	0,053	0,079

Fonte: Autoria Própria

O Quadro 30 mostra a diferença das medidas de avaliação para a base *Enron* completa (Quadro 9) em relação a base *Enron* que teve 30% dos valores imputados (Quadro 29). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 30 - Diferença de avaliação *Enron* Original e 30% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,089	-0,040	0,053	0,003	0,010	0,006
	Média	0,090	-0,041	0,056	0,009	0,010	0,006
	Mediana	0,089	-0,040	0,053	0,003	0,010	0,006
	KNN	0,075	-0,028	0,009	0,014	0,004	0,005
HL	Moda	-0,014	0,005	-0,005	0,000	-0,002	0,021
	Média	-0,014	0,005	-0,005	0,000	-0,002	0,021
	Mediana	-0,014	0,005	-0,005	0,000	-0,002	0,021
	KNN	-0,013	0,003	0,000	-0,002	-0,002	0,004
Medida F	Moda	0,087	-0,040	0,052	0,015	0,020	0,001
	Média	0,088	-0,041	0,054	0,019	0,020	0,001
	Mediana	0,087	-0,040	0,052	0,015	0,020	0,001
	KNN	0,069	-0,024	0,006	0,023	0,008	0,006

Fonte: Autoria Própria

O Quadro 31 mostra quais foram os melhores algoritmos de imputação para cada classificador aplicado na base *Enron* 30% Imputada, de acordo com o Quadro 30.

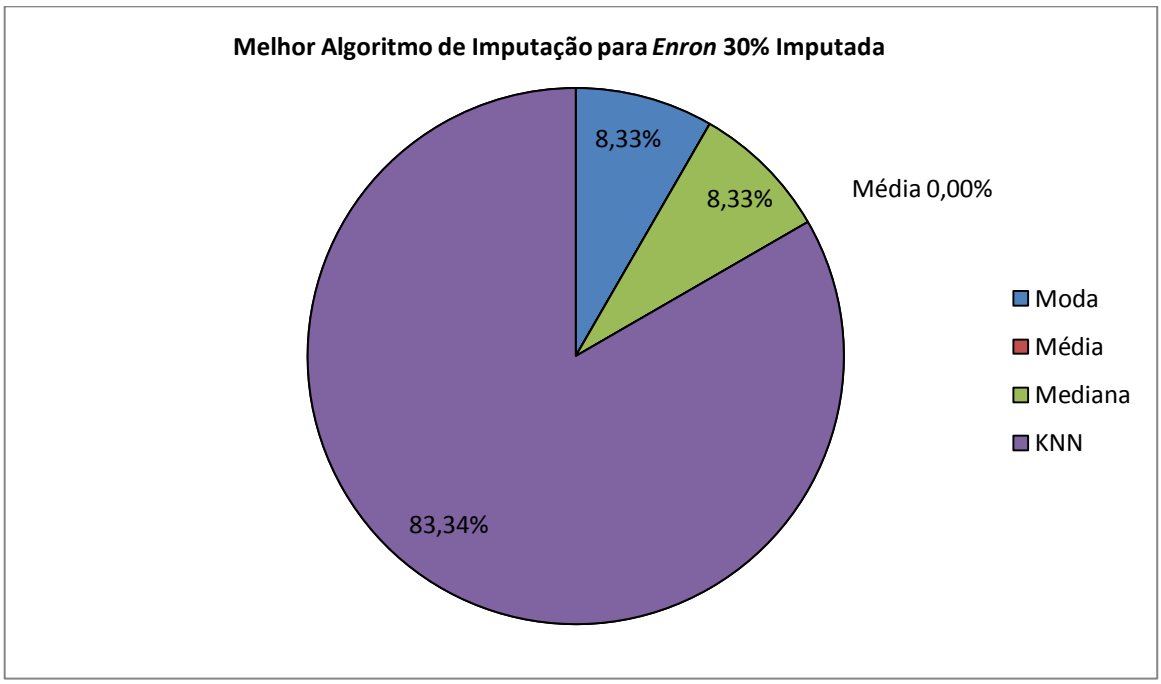
Quadro 31 - Melhores algoritmos de imputação para classificadores da *Enron* 30% Imputada

Classificador	Algoritmo de Imputação
BR com J48	KNN
BR com NB	KNN
LP com J48	KNN
LP com NB	Moda, Mediana
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 31), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Enron* 30% Imputada, conforme é mostrado no Gráfico 7.

Gráfico 7 - Melhor algoritmo de imputação para *Enron* 30% Imputada



Fonte: Autoria Própria

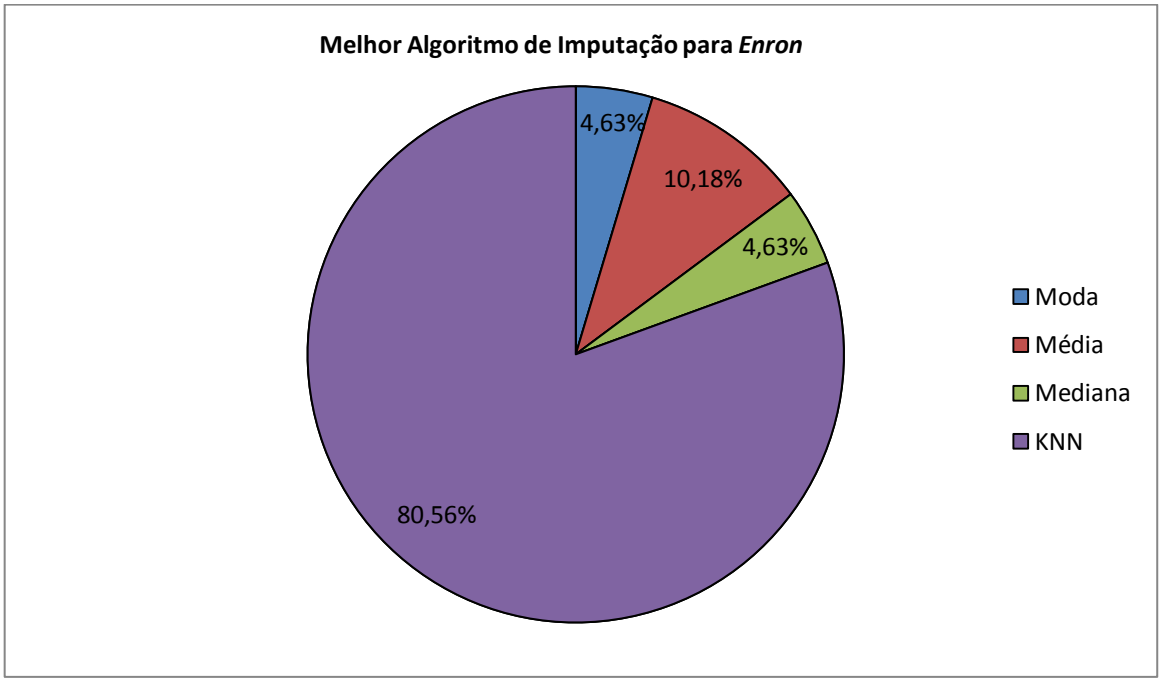
Pelo Gráfico 7 é possível observar que o algoritmo Imputação KNN Iterativo obteve a maioria dos resultados mais próximos aos da base *Enron* Original,

representando 83,34% dos melhores resultados. Por isso, é considerado o melhor algoritmo de imputação para a base *Enron* 30% Imputada. Depois do KNN Iterativo, aparecem as imputações pela Moda e pela Mediana como melhores algoritmos de imputação para a *Enron* 30% Imputada, pois possuem o mesmo percentual de melhores resultados, sendo 8,33%. O pior resultado é da Imputação pela Média que representa 0% dos resultados que mais se aproximaram da base de dados original.

4.2.2.4 Melhor algoritmo de imputação para *Enron*

A partir dos Gráficos 5, 6 e 7 pode-se calcular a média das porcentagens dos melhores algoritmos de imputação para as bases de dados *Enron* com 10%, 20% e 30% de valores imputados e obter o melhor algoritmo de imputação para as bases de dados *Enron*, conforme é demonstrado no Gráfico 8.

Gráfico 8 - Melhor algoritmo de imputação para *Enron*



Fonte: Autoria Própria

O Gráfico 8 demonstra que o algoritmo Imputação KNN Iterativo obteve grande parte dos resultados mais próximos aos da base *Enron* Original, representando 80,56% dos melhores resultados. Então, é considerado o melhor algoritmo de imputação para as bases *Enron*. Imputação pela Média é o segundo melhor algoritmo de imputação para as bases *Enron* que foram imputadas com

10,18% dos melhores resultados. Os algoritmos Imputação pela Moda e Imputação pela Mediana alcançaram os piores resultados, cada um representando apenas 4,63% dos melhores resultados alcançados.

4.2.3 Base de Dados *Mediamill*

Foram realizados experimentos com três porcentagens diferentes de valores imputados na base de dados *Mediamill*, por isso os resultados serão apresentados separadamente para cada porcentagem.

4.2.3.1 *Mediamill* 10% Imputada

O Quadro 32 mostra os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Mediamill* com 10% de seus valores imputados

Quadro 32 - Avaliação *Mediamill* 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,059	0,210	0,321	0,315	0,118	0,067
	Média	0,064	0,213	0,324	0,314	0,116	0,068
	Mediana	0,063	0,208	0,317	0,313	0,114	0,067
	KNN	0,068	0,192	0,366	0,309	0,128	0,066
HL	Moda	0,366	0,093	0,048	0,053	0,046	0,104
	Média	0,336	0,091	0,047	0,053	0,046	0,104
	Mediana	0,345	0,094	0,048	0,053	0,046	0,104
	KNN	0,327	0,105	0,044	0,054	0,044	0,105
Medida F	Moda	0,113	0,310	0,439	0,458	0,163	0,125
	Média	0,121	0,314	0,441	0,456	0,162	0,126
	Mediana	0,120	0,308	0,435	0,457	0,161	0,125
	KNN	0,128	0,289	0,480	0,453	0,177	0,124

Fonte: Autoria Própria

O Quadro 33 mostra a diferença das medidas de avaliação para a base *Mediamill* completa (Quadro 10) em relação a base *Mediamill* que teve 10% dos

valores imputados (Quadro 32). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 33 - Diferença de avaliação *Mediamill* Original e 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,013	-0,017	0,057	-0,005	0,011	0,000
	Média	0,008	-0,020	0,054	-0,004	0,013	-0,001
	Mediana	0,009	-0,015	0,061	-0,003	0,015	0,000
	KNN	0,004	0,001	0,012	0,001	0,001	0,001
HL	Moda	-0,053	0,012	-0,005	0,001	-0,002	0,001
	Média	-0,023	0,014	-0,004	0,001	-0,002	0,001
	Mediana	-0,032	0,011	-0,005	0,001	-0,002	0,001
	KNN	-0,014	0,000	-0,001	0,000	0,000	0,000
Medida F	Moda	0,022	-0,021	0,049	-0,004	0,016	-0,001
	Média	0,014	-0,025	0,047	-0,002	0,017	-0,002
	Mediana	0,015	-0,019	0,053	-0,003	0,018	-0,001
	KNN	0,007	0,000	0,008	0,001	0,002	0,000

Fonte: Autoria Própria

O Quadro 34 se refere a base *Mediamill* 10% Imputada e mostra os melhores algoritmos de imputação para cada classificador, de acordo com o Quadro 33.

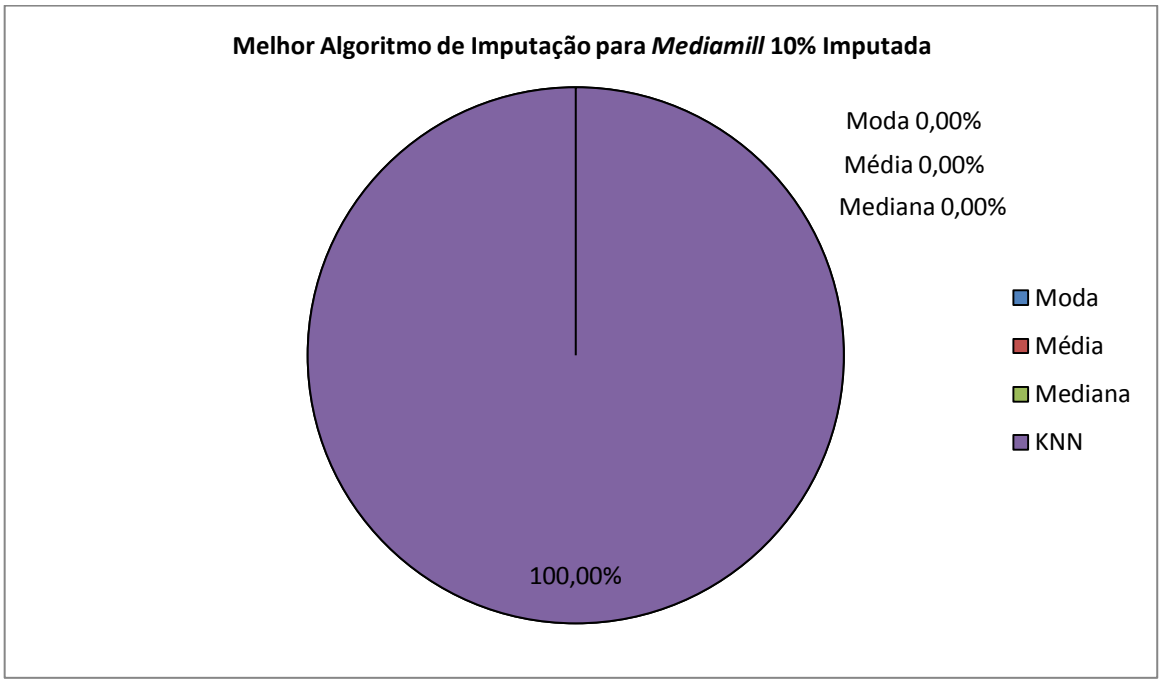
Quadro 34 - Melhores algoritmos de imputação para classificadores da *Mediamill* 10% Imputada

Classificador	Algoritmo de Imputação
BR com J48	KNN
BR com NB	KNN
LP com J48	KNN
LP com NB	KNN
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 34), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Mediamill* 10% Imputada, conforme é mostrado no Gráfico 9.

Gráfico 9 - Melhor algoritmo de imputação para *Mediamill* 10% Imputada



Fonte: Autoria Própria

O Gráfico 9 prova que o algoritmo Imputação KNN Iterativo é o melhor para a base *Mediamill* 10% Imputada, pois totalizou 100% dos melhores resultados. Os demais algoritmos de imputação não foram considerados como melhores para nenhum classificador aplicado, por isso correspondem a 0% do total.

4.2.3.2 *Mediamill* 20% Imputada

O Quadro 35 apresenta os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Mediamill* com 20% de seus valores imputados.

Quadro 35 - Avaliação *Mediamill* 20% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,054	0,231	0,290	0,313	0,109	0,069
	Média	0,048	0,237	0,298	0,311	0,109	0,070
	Mediana	0,050	0,225	0,291	0,312	0,110	0,069
	KNN	0,060	0,194	0,343	0,307	0,122	0,066
HL	Moda	0,395	0,083	0,050	0,053	0,046	0,102
	Média	0,454	0,080	0,050	0,053	0,046	0,101
	Mediana	0,436	0,085	0,050	0,053	0,046	0,103
	KNN	0,368	0,104	0,046	0,054	0,045	0,105
Medida F	Moda	0,104	0,332	0,410	0,455	0,152	0,128
	Média	0,093	0,339	0,417	0,453	0,153	0,129
	Mediana	0,096	0,327	0,409	0,454	0,155	0,127
	KNN	0,114	0,290	0,457	0,450	0,169	0,124

Fonte: Autoria Própria

O Quadro 36 mostra a diferença das medidas de avaliação para a base *Mediamill* completa (Quadro 10) em relação a base *Mediamill* que teve 20% dos valores imputados (Quadro 35). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 36 - Diferença de avaliação *Mediamill* Original e 20% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,018	-0,038	0,088	-0,003	0,020	-0,002
	Média	0,024	-0,044	0,080	-0,001	0,020	-0,003
	Mediana	0,022	-0,032	0,087	-0,002	0,019	-0,002
	KNN	0,012	-0,001	0,035	0,003	0,007	0,001
HL	Moda	-0,082	0,022	-0,007	0,001	-0,002	0,003
	Média	-0,141	0,025	-0,007	0,001	-0,002	0,004
	Mediana	-0,123	0,020	-0,007	0,001	-0,002	0,002
	KNN	-0,055	0,001	-0,003	0,000	-0,001	0,000
Medida F	Moda	0,031	-0,043	0,078	-0,001	0,027	-0,004
	Média	0,042	-0,050	0,071	0,001	0,026	-0,005
	Mediana	0,039	-0,038	0,079	0,000	0,024	-0,003
	KNN	0,021	-0,001	0,031	0,004	0,010	0,000

Fonte: Autoria Própria

O Quadro 37 mostra quais foram os melhores algoritmos de imputação para cada classificador aplicado na base *Mediamill* 20% Imputada, de acordo com o Quadro 36.

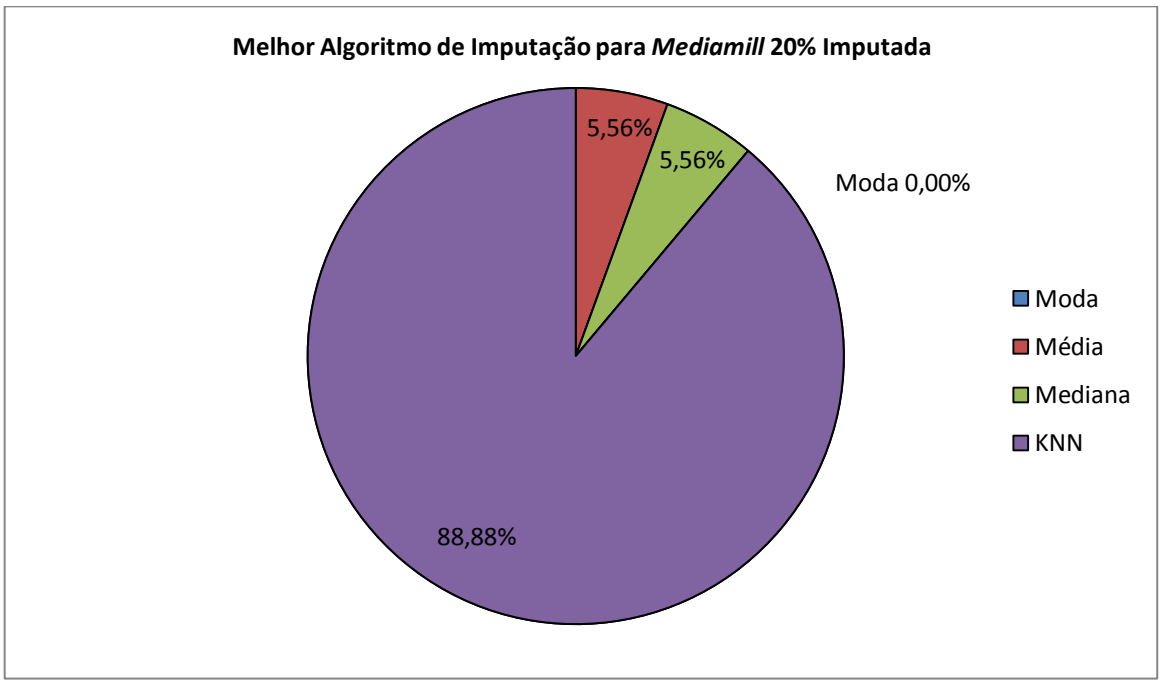
Quadro 37 - Melhores algoritmos de imputação para classificadores da *Mediamill* 20% Imputada

Classificador	Algoritmo de Imputação
BR com J48	KNN
BR com NB	KNN
LP com J48	KNN
LP com NB	Média, Mediana, KNN
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 37), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Mediamill* 20% Imputada, conforme é mostrado no Gráfico 10.

Gráfico 10 - Melhor algoritmo de imputação para *Mediamill* 20% Imputada



Fonte: Autoria Própria

No Gráfico 10 é possível verificar que o algoritmo Imputação KNN Iterativo foi o melhor para a base *Mediamill* 20% Imputada, pois obteve 88,88% dos melhores

resultados. O segundo melhor resultado foi da Imputação pela Média e também da Imputação pela Mediana, cada uma representando 5,56% dos resultados mais próximos aos reais. O pior algoritmo para a base *Medical* 20% Imputada foi Imputação pela Moda equivalendo a 0% dos melhores resultados.

4.2.3.3 *Mediamill* 30% Imputada

O Quadro 38 mostra os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Mediamill* com 30% de seus valores imputados.

Quadro 38 - Avaliação *Mediamill* 30% Imputada

	Algoritmos de Imputação	**BR**		**LP**		**RAKEL**	
		J48	**NB**	**J48**	**NB**	**J48**	**NB**
Acurácia	**Moda**	0,048	0,246	0,279	0,310	0,106	0,072
	Média	0,052	0,263	0,276	0,309	0,103	0,074
	Mediana	0,046	0,244	0,275	0,311	0,103	0,072
	KNN	0,054	0,197	0,319	0,306	0,111	0,067
HL	**Moda**	0,443	0,076	0,051	0,053	0,047	0,102
	Média	0,405	0,071	0,051	0,053	0,047	0,100
	Mediana	0,462	0,077	0,051	0,053	0,047	0,102
	KNN	0,402	0,101	0,047	0,054	0,046	0,105
Medida F	**Moda**	0,092	0,348	0,401	0,450	0,150	0,133
	Média	0,099	0,364	0,396	0,449	0,144	0,134
	Mediana	0,090	0,346	0,396	0,451	0,144	0,132
	KNN	0,104	0,295	0,436	0,449	0,152	0,124

Fonte: Autoria Própria

O Quadro 39 mostra a diferença das medidas de avaliação para a base *Mediamill* completa (Quadro 10) em relação a base *Mediamill* que teve 30% dos valores imputados (Quadro 38). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 39 - Diferença de avaliação *Mediamill* Original e 30% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,024	-0,053	0,099	0,000	0,023	-0,005
	Média	0,020	-0,070	0,102	0,001	0,026	-0,007
	Mediana	0,026	-0,051	0,103	-0,001	0,026	-0,005
	KNN	0,018	-0,004	0,059	0,004	0,018	0,000
HL	Moda	-0,130	0,029	-0,008	0,001	-0,003	0,003
	Média	-0,092	0,034	-0,008	0,001	-0,003	0,005
	Mediana	-0,149	0,028	-0,008	0,001	-0,003	0,003
	KNN	-0,089	0,004	-0,004	0,000	-0,002	0,000
Medida F	Moda	0,043	-0,059	0,087	0,004	0,029	-0,009
	Média	0,036	-0,075	0,092	0,005	0,035	-0,010
	Mediana	0,045	-0,057	0,092	0,003	0,035	-0,008
	KNN	0,031	-0,006	0,052	0,005	0,027	0,000

Fonte: Autoria Própria

O Quadro 40 se refere a base *Mediamill* 30% Imputada e mostra os melhores algoritmos de imputação para cada classificador aplicado, de acordo com o Quadro 39.

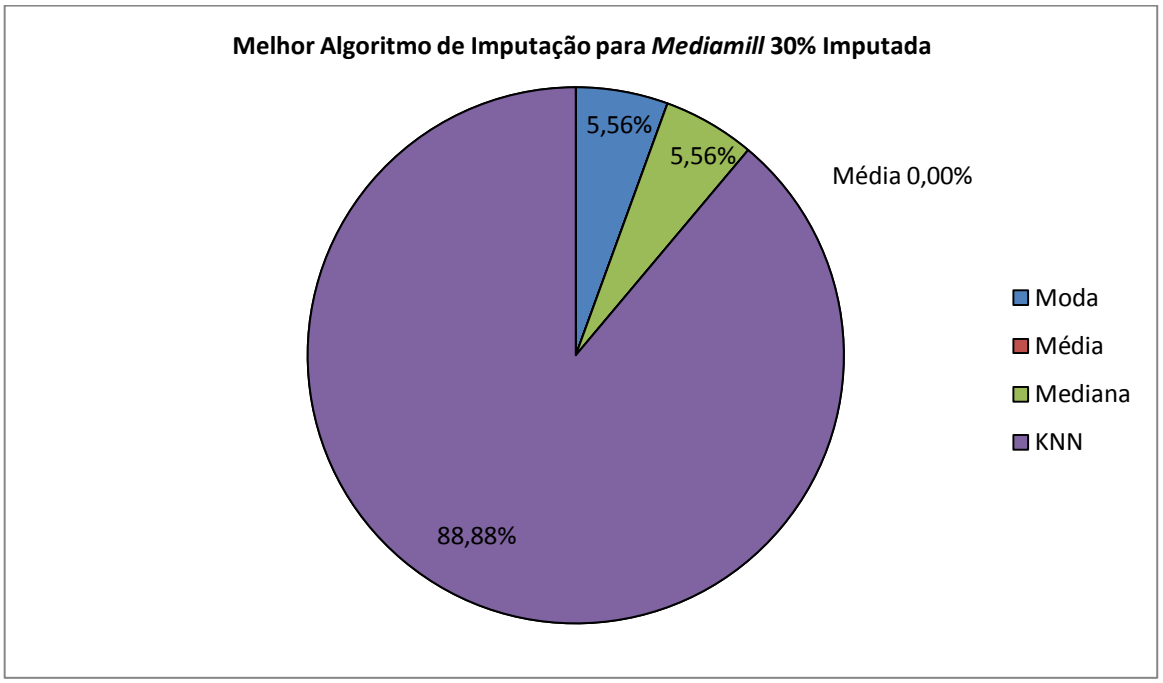
Quadro 40 - Melhores algoritmos de imputação para classificadores da *Mediamill* 30% Imputada

Classificador	Algoritmo de Imputação
BR com J48	KNN
BR com NB	KNN
LP com J48	KNN
LP com NB	Moda, Mediana, KNN
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 40), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Mediamill* 30% Imputada, conforme pode ser visualizado no Gráfico 11.

Gráfico 11 - Melhor algoritmo de imputação para *Mediamill* 30% Imputada



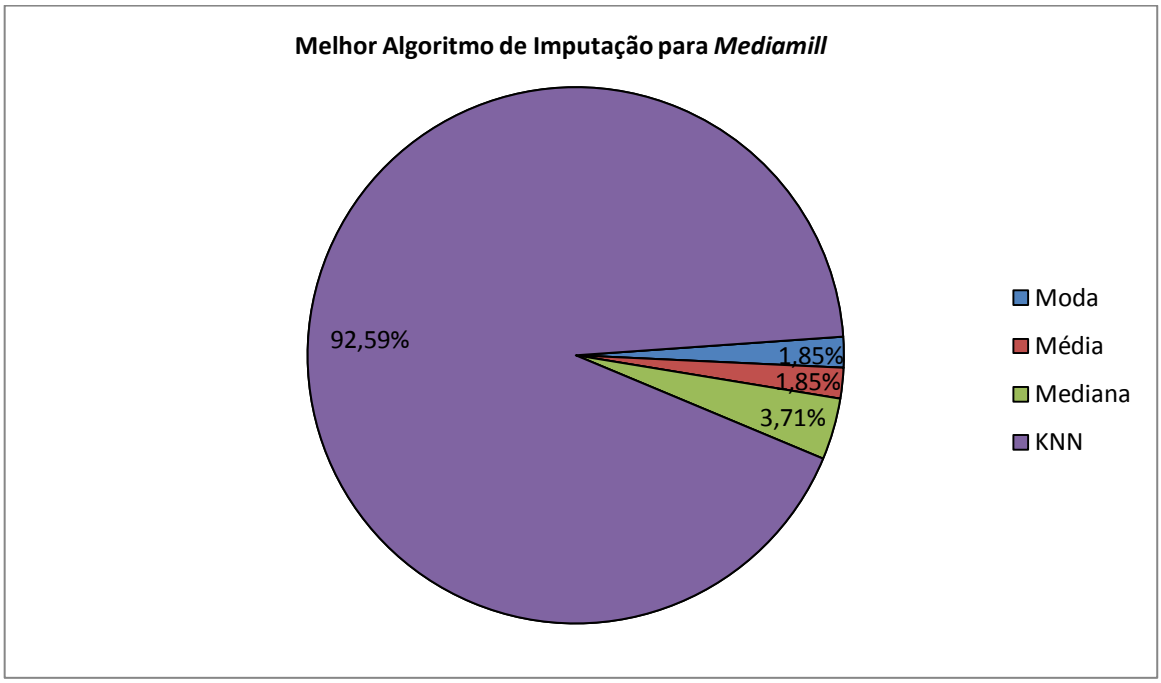
Fonte: Autoria Própria

O Gráfico 11 demonstra que o algoritmo Imputação KNN Iterativo é o melhor para a base *Mediamill* 30% Imputada, pois obteve 88,88% dos melhores resultados. As Imputações pela Moda e Mediana representam o segundo melhor algoritmo para a base *Mediamill* 30% Imputada, pois cada uma representa 5,56% de valores mais próximos aos da base *Mediamill* Original. O pior resultado ficou com o algoritmo Imputação pela Média correspondendo a 0% do total de melhores resultados.

4.2.3.4 Melhor algoritmo de imputação para *Mediamill*

A partir dos Gráficos 9, 10 e 11 pode-se calcular a média das porcentagens dos melhores algoritmos de imputação para as bases de dados *Mediamill* com 10%, 20% e 30% de valores imputados e obter o melhor algoritmo de imputação para as bases de dados *Mediamill*, conforme é mostrado no Gráfico 12.

Gráfico 12 - Melhor algoritmo de imputação para *Mediamill*



Fonte: Autoria Própria

O Gráfico 12 revela que o algoritmo Imputação KNN Iterativo obteve os resultados mais próximos aos da base *Mediamill* Original, representando 92,59%. Por isso, é o melhor algoritmo de imputação para as bases *Mediamill*. Imputação pela Mediana é o segundo melhor algoritmo para as bases *Mediamill* possuindo 3,71% dos melhores resultados. Os algoritmos Imputação pela Moda e Imputação pela Média foram os piores, cada um correspondendo a apenas 1,85% do total.

4.2.4 Base de Dados *Medical*

Foram realizados experimentos com três porcentagens diferentes de valores imputados na base de dados *Medical*, por isso os resultados serão apresentados separadamente para cada porcentagem.

4.2.4.1 *Medical* 10% Imputada

O Quadro 41 mostra os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os

classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Medical* com 10% de seus valores imputados.

Quadro 41 - Avaliação *Medical* 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,449	0,353	0,665	0,391	0,411	0,211
	Média	0,449	0,350	0,659	0,391	0,411	0,217
	Mediana	0,449	0,353	0,665	0,391	0,411	0,211
	KNN	0,642	0,349	0,673	0,394	0,412	0,225
HL	Moda	0,030	0,030	0,017	0,029	0,019	0,025
	Média	0,030	0,030	0,017	0,029	0,019	0,025
	Mediana	0,030	0,030	0,017	0,029	0,019	0,025
	KNN	0,016	0,030	0,016	0,029	0,019	0,025
Medida F	Moda	0,580	0,455	0,690	0,418	0,544	0,328
	Média	0,580	0,455	0,685	0,418	0,544	0,334
	Mediana	0,580	0,455	0,690	0,418	0,544	0,328
	KNN	0,719	0,455	0,698	0,420	0,542	0,342

Fonte: Autoria Própria

O Quadro 42 mostra a diferença das medidas de avaliação para a base *Medical* completa (Quadro 11) em relação à base *Medical* 10% Imputada (Quadro 41).

Quadro 42 - Diferença de avaliação *Medical* Original e 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,308	0,009	0,068	0,005	0,055	0,031
	Média	0,308	0,012	0,074	0,005	0,055	0,025
	Mediana	0,308	0,009	0,068	0,005	0,055	0,031
	KNN	0,115	0,013	0,060	0,002	0,054	0,017
HL	Moda	-0,019	-0,001	-0,003	0,000	-0,001	0,000
	Média	-0,019	-0,001	-0,003	0,000	-0,001	0,000
	Mediana	-0,019	-0,001	-0,003	0,000	-0,001	0,000
	KNN	-0,005	-0,001	-0,002	0,000	-0,001	0,000
Medida F	Moda	0,229	0,015	0,063	0,002	0,040	0,038
	Média	0,229	0,015	0,068	0,002	0,040	0,032
	Mediana	0,229	0,015	0,063	0,002	0,040	0,038
	KNN	0,090	0,015	0,055	0,000	0,042	0,024

Fonte: Autoria Própria

O Quadro 43 se refere a base *Medical* 10% Imputada e mostra os melhores algoritmos de imputação para cada classificador, de acordo com o Quadro 42.

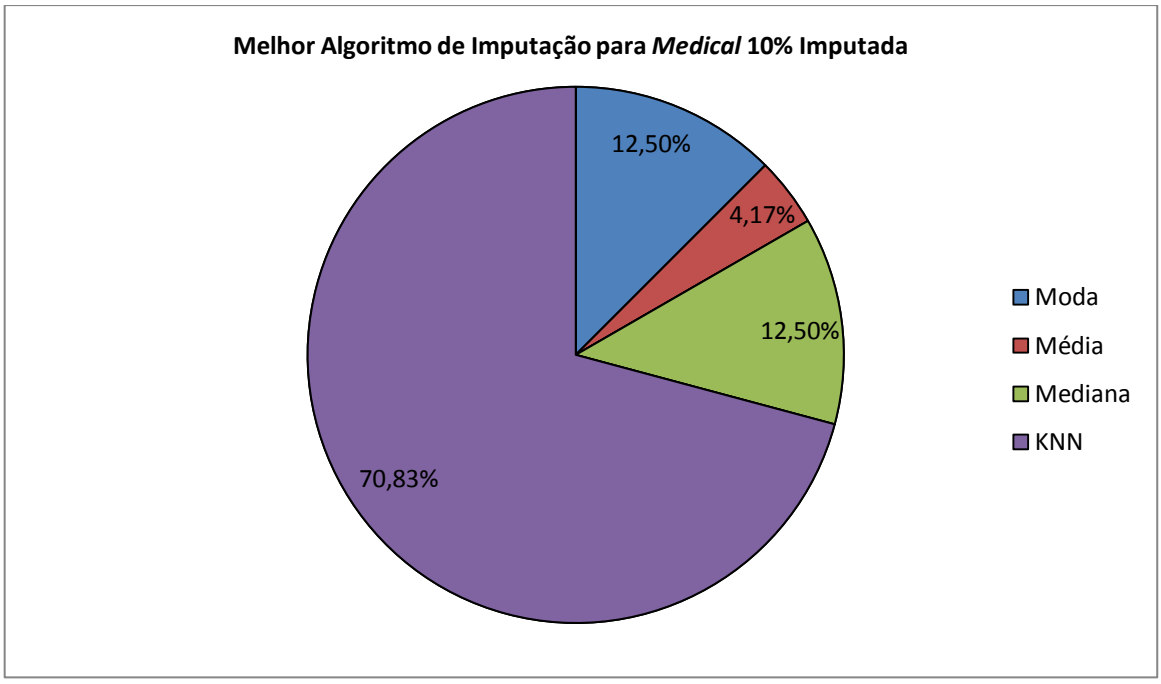
Quadro 43 - Melhores algoritmos de imputação para classificadores da *Medical* 10% Imputada

Classificador	Algoritmo de Imputação
BR com J48	KNN
BR com NB	Moda, Mediana
LP com J48	KNN
LP com NB	KNN
RAKEL com J48	Moda, Média, Mediana, KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 43), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Medical* 10% Imputada, conforme é mostrado no Gráfico 13.

Gráfico 13 - Melhor algoritmo de imputação para *Medical* 10% Imputada



Fonte: Autoria Própria

No Gráfico 13 é possível constatar que o algoritmo Imputação KNN Iterativo foi o melhor para a base *Medical* 10% Imputada, representando 70,83% dos resultados mais próximos aos reais. Depois do KNN Iterativo, aparecem as

imputações pela Moda e pela Mediana como melhores algoritmos de imputação para a *Medical* 10% Imputada, pois possuem o mesmo percentual de melhores resultados: 12,5%. O pior algoritmo para a base foi o Imputação pela Média que teve apenas 4,17% dos resultados que mais se aproximaram da base *Medical* Original.

4.2.4.2 *Medical* 20% Imputada

O Quadro 44 apresenta os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Medical* com 20% de seus valores imputados.

Quadro 44 - Avaliação *Medical* 20% Imputada

	Algoritmos de Imputação	**BR**		**LP**		**RAKEL**	
		J48	**NB**	**J48**	**NB**	**J48**	**NB**
Acurácia	**Moda**	0,452	0,350	0,625	0,372	0,383	0,196
	Média	0,454	0,348	0,634	0,370	0,383	0,188
	Mediana	0,452	0,350	0,625	0,372	0,383	0,196
	KNN	0,418	0,354	0,632	0,375	0,370	0,205
HL	**Moda**	0,030	0,031	0,018	0,030	0,020	0,025
	Média	0,030	0,031	0,018	0,030	0,020	0,025
	Mediana	0,030	0,031	0,018	0,030	0,020	0,025
	KNN	0,032	0,030	0,018	0,030	0,020	0,025
Medida F	**Moda**	0,565	0,445	0,655	0,397	0,503	0,309
	Média	0,567	0,445	0,665	0,394	0,503	0,300
	Mediana	0,565	0,445	0,655	0,397	0,503	0,309
	KNN	0,548	0,450	0,660	0,399	0,491	0,319

Fonte: Autoria Própria

O Quadro 45 mostra a diferença das medidas de avaliação para a base *Medical* completa (Quadro 11) em relação a base *Medical* que teve 20% dos valores imputados (Quadro 44). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 45 - Diferença de avaliação *Medical* Original e 20% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,305	0,012	0,108	0,024	0,083	0,046
	Média	0,303	0,014	0,099	0,026	0,083	0,054
	Mediana	0,305	0,012	0,108	0,024	0,083	0,046
	KNN	0,339	0,008	0,101	0,021	0,096	0,037
HL	Moda	-0,019	-0,002	-0,004	-0,001	-0,002	0,000
	Média	-0,019	-0,002	-0,004	-0,001	-0,002	0,000
	Mediana	-0,019	-0,002	-0,004	-0,001	-0,002	0,000
	KNN	-0,021	-0,001	-0,004	-0,001	-0,002	0,000
Medida F	Moda	0,244	0,025	0,098	0,023	0,081	0,057
	Média	0,242	0,025	0,088	0,026	0,081	0,066
	Mediana	0,244	0,025	0,098	0,023	0,081	0,057
	KNN	0,261	0,020	0,093	0,021	0,093	0,047

Fonte: Autoria Própria

O Quadro 46 mostra quais foram os melhores algoritmos de imputação para cada classificador aplicado na base *Medical* 20% Imputada, de acordo com o Quadro 45.

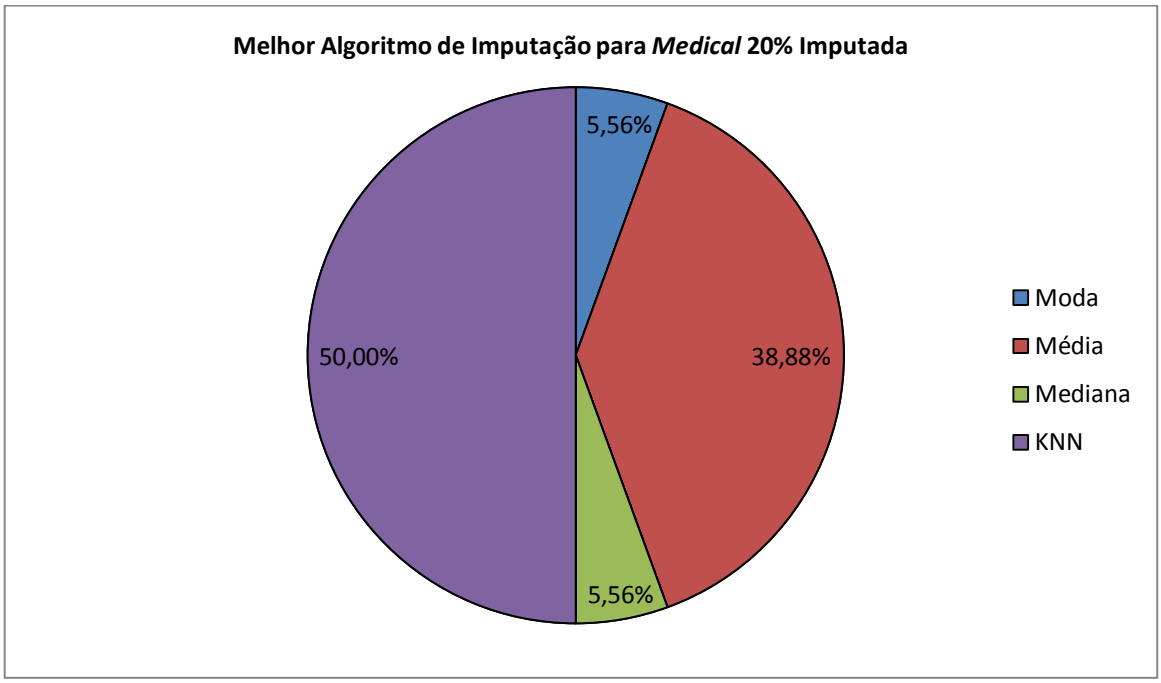
Quadro 46 - Melhores algoritmos de imputação para classificadores da *Medical* 20% Imputada

Classificador	Algoritmo de Imputação
BR com J48	Média
BR com NB	KNN
LP com J48	Média
LP com NB	KNN
RAKEL com J48	Moda, Média, Mediana
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 46), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Medical* 20% Imputada, conforme é mostrado no Gráfico 14.

Gráfico 14 - Melhor algoritmo de imputação para *Medical* 20% Imputada



Fonte: Autoria Própria

No Gráfico 14 é possível verificar que o algoritmo Imputação KNN Iterativo foi o melhor para a base *Medical* 20% Imputada, pois obteve 50% dos melhores resultados. O segundo melhor resultado é da Imputação pela Média representando 38,88% dos resultados mais próximos aos reais. Os piores algoritmos para a base *Medical* 20% Imputada foram Imputação pela Moda e Imputação pela Mediana, pois cada um conseguiu somente 5,56% do total.

4.2.4.3 *Medical* 30% Imputada

O Quadro 47 mostra os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Medical* com 30% de seus valores imputados.

Quadro 47 - Avaliação *Medical* 30% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,423	0,346	0,543	0,368	0,319	0,159
	Média	0,420	0,345	0,539	0,360	0,319	0,156
	Mediana	0,423	0,346	0,543	0,368	0,319	0,159
	KNN	0,423	0,346	0,535	0,377	0,320	0,168
HL	Moda	0,030	0,031	0,023	0,030	0,021	0,025
	Média	0,031	0,031	0,023	0,030	0,021	0,026
	Mediana	0,030	0,031	0,023	0,030	0,021	0,025
	KNN	0,031	0,031	0,023	0,030	0,021	0,025
Medida F	Moda	0,538	0,438	0,573	0,395	0,448	0,270
	Média	0,532	0,435	0,566	0,389	0,448	0,265
	Mediana	0,538	0,438	0,573	0,395	0,448	0,270
	KNN	0,534	0,440	0,559	0,403	0,450	0,283

Fonte: Autoria Própria

O Quadro 48 mostra a diferença das medidas de avaliação para a base *Medical* completa (Quadro 11) em relação a base *Medical* que teve 30% dos valores imputados (Quadro 47). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 48 - Diferença de avaliação *Medical* Original e 30% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,334	0,016	0,190	0,028	0,147	0,083
	Média	0,337	0,017	0,194	0,036	0,147	0,086
	Mediana	0,334	0,016	0,190	0,028	0,147	0,083
	KNN	0,334	0,016	0,198	0,019	0,146	0,074
HL	Moda	-0,019	-0,002	-0,009	-0,001	-0,003	0,000
	Média	-0,020	-0,002	-0,009	-0,001	-0,003	-0,001
	Mediana	-0,019	-0,002	-0,009	-0,001	-0,003	0,000
	KNN	-0,020	-0,002	-0,009	-0,001	-0,003	0,000
Medida F	Moda	0,271	0,032	0,180	0,025	0,136	0,096
	Média	0,277	0,035	0,187	0,031	0,136	0,101
	Mediana	0,271	0,032	0,180	0,025	0,136	0,096
	KNN	0,275	0,030	0,194	0,017	0,134	0,083

Fonte: Autoria Própria

O Quadro 49 se refere a base Medical 30% Imputada e mostra os melhores algoritmos de imputação para cada classificador aplicado, de acordo com o Quadro 48.

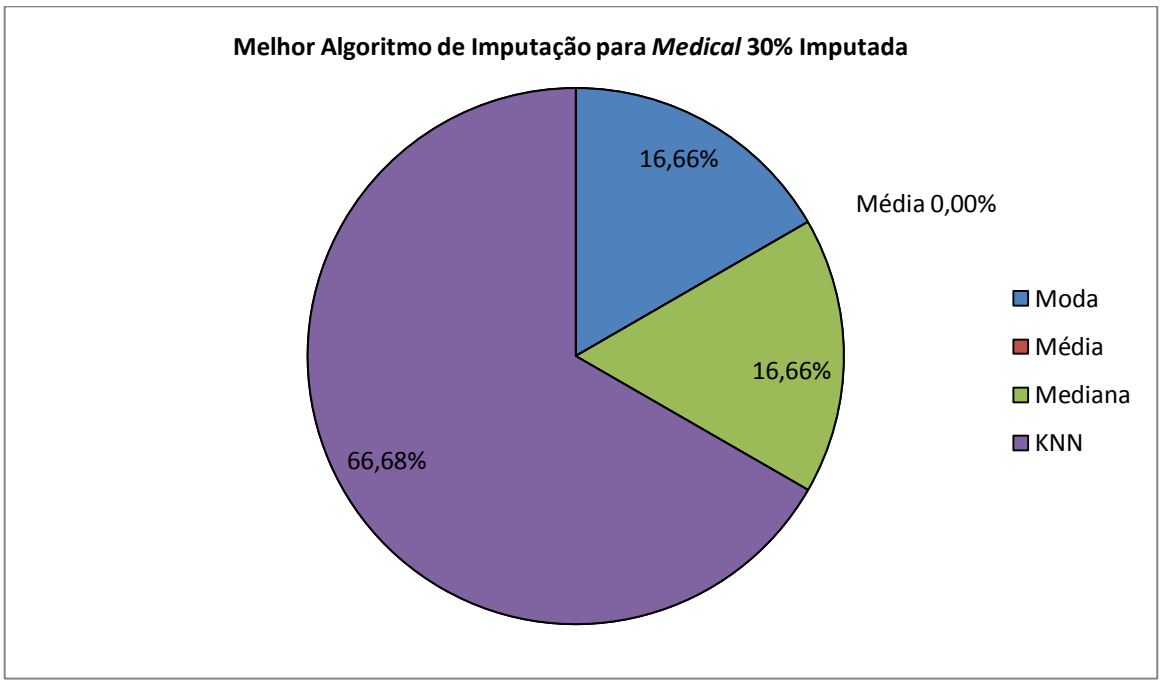
Quadro 49 - Melhores algoritmos de imputação para classificadores da Medical 30% Imputada

Classificador	Algoritmo de Imputação
BR com J48	Moda, Mediana
BR com NB	KNN
LP com J48	Moda, Mediana
LP com NB	KNN
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 49), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Medical* 30% Imputada, conforme pode ser visualizado no Gráfico 15.

Gráfico 15 - Melhor algoritmo de imputação para *Medical* 30% Imputada



Fonte: Autoria Própria

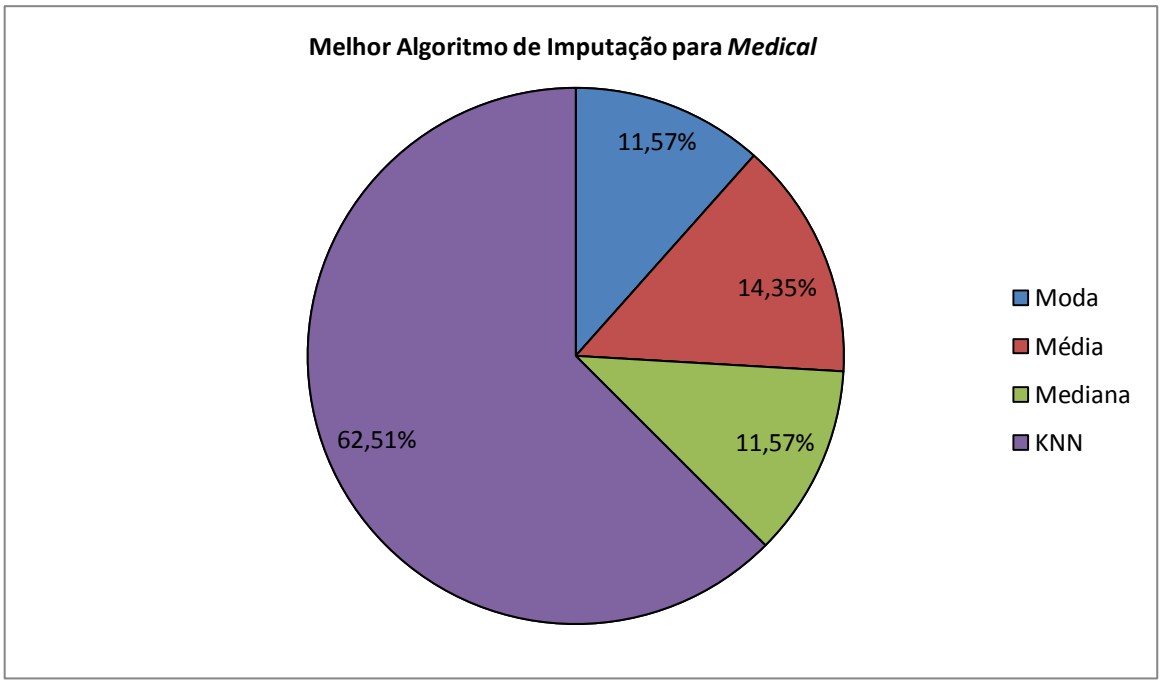
O Gráfico 15 demonstra que o algoritmo Imputação KNN Iterativo é o melhor para a base *Medical* 30% Imputada, pois obteve 66,68% dos melhores resultados.

As Imputações pela Moda e Mediana representam o segundo melhor resultado para a *Medical* 30% Imputada com 16,66% de valores mais próximos aos da base *Medical* Original. O pior resultado ficou com o algoritmo Imputação pela Média correspondendo a 0% do total de melhores resultados.

4.2.4.4 Melhor algoritmo de imputação para *Medical*

A partir dos Gráficos 13, 14 e 15 pode-se calcular a média das porcentagens dos melhores algoritmos de imputação para as bases de dados *Medical* com 10%, 20% e 30% de valores imputados e obter o melhor algoritmo de imputação para as bases de dados *Medical*, conforme é apresentado no Gráfico 16.

Gráfico 16 - Melhor algoritmo de imputação para *Medical*



Fonte: Autoria Própria

O Gráfico 16 revela que o algoritmo Imputação KNN Iterativo obteve a maioria dos resultados mais próximos aos da base *Medical* Original, representando 62,51% dos melhores resultados. Desta maneira, é considerado o melhor algoritmo de imputação para as bases *Medical*. Imputação pela Média foi o segundo melhor algoritmo para as bases *Medical*, possuindo 14,35% dos melhores resultados. Os algoritmos Imputação pela Moda e Imputação pela Mediana alcançaram os piores

resultados, cada um correspondendo a 11,57% dos resultados que mais se aproximaram dos valores reais.

4.2.5 Base de Dados *Scene*

Foram realizados experimentos com três porcentagens diferentes de valores imputados na base de dados *Scene*, por isso os resultados serão apresentados separadamente para cada porcentagem.

4.2.5.1 *Scene* 10% Imputada

O Quadro 50 apresenta os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Scene* com 10% de seus valores imputados.

Quadro 50 - Avaliação *Scene* 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,465	0,433	0,496	0,614	0,553	0,569
	Média	0,493	0,460	0,551	0,627	0,586	0,598
	Mediana	0,482	0,454	0,535	0,621	0,584	0,594
	KNN	0,499	0,466	0,579	0,633	0,613	0,603
HL	Moda	0,160	0,156	0,178	0,135	0,166	0,155
	Média	0,144	0,179	0,156	0,132	0,148	0,144
	Mediana	0,147	0,182	0,163	0,134	0,145	0,147
	KNN	0,155	0,181	0,147	0,130	0,138	0,143
Medida F	Moda	0,556	0,566	0,508	0,629	0,615	0,624
	Média	0,593	0,585	0,565	0,643	0,641	0,653
	Mediana	0,589	0,580	0,545	0,639	0,644	0,648
	KNN	0,580	0,587	0,589	0,653	0,662	0,656

Fonte: Autoria Própria

O Quadro 51 mostra a diferença das medidas de avaliação para a base *Scene* completa (Quadro 12) em relação a base *Scene* que teve 10% dos valores

imputados (Quadro 50). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 51 - Diferença de avaliação *Scene* Original e 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,045	0,037	0,079	0,009	0,048	0,034
	Média	0,017	0,010	0,024	-0,004	0,015	0,005
	Mediana	0,028	0,016	0,040	0,002	0,017	0,009
	KNN	0,011	0,004	-0,004	-0,010	-0,012	0,000
HL	Moda	-0,009	0,022	-0,029	0,000	-0,020	-0,010
	Média	0,007	-0,001	-0,007	0,003	-0,002	0,001
	Mediana	0,004	-0,004	-0,014	0,001	0,001	-0,002
	KNN	-0,004	-0,003	0,002	0,005	0,008	0,002
Medida F	Moda	0,040	0,023	0,079	0,010	0,033	0,030
	Média	0,003	0,004	0,022	-0,004	0,007	0,001
	Mediana	0,007	0,009	0,042	0,000	0,004	0,006
	KNN	0,016	0,002	-0,002	-0,014	-0,014	-0,002

Fonte: Autoria Própria

O Quadro 52 mostra quais foram os melhores algoritmos de imputação para cada classificador aplicado na base *Scene* 10% Imputada, de acordo com o Quadro 51.

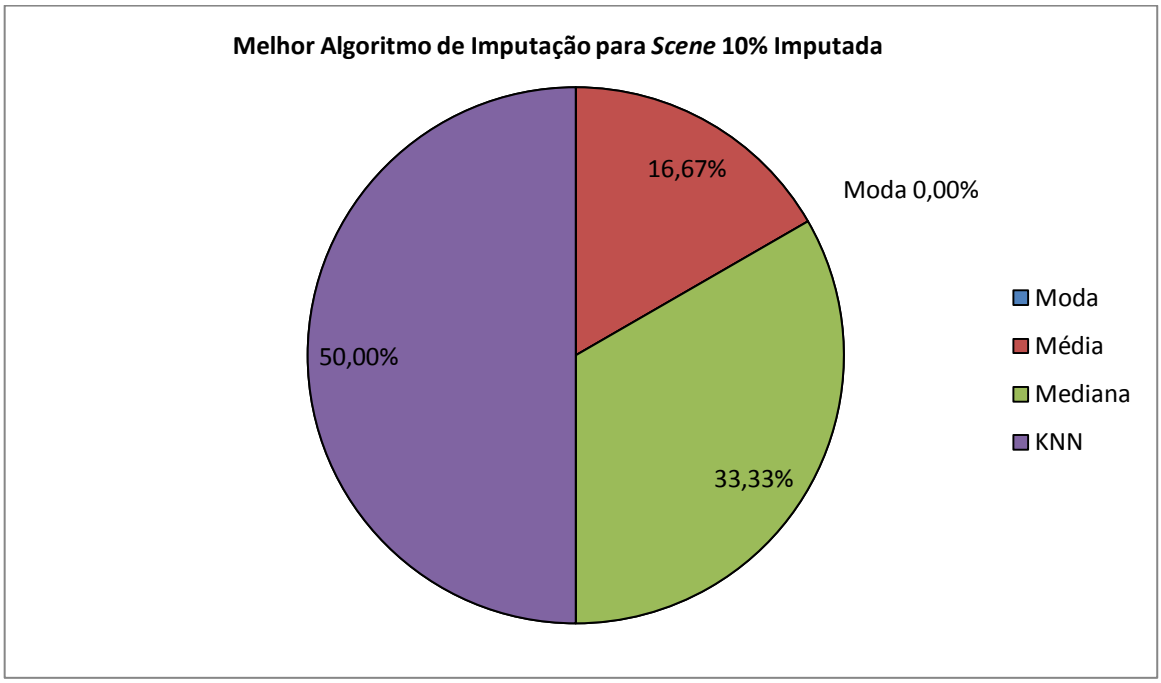
Quadro 52 - Melhores algoritmos de imputação para classificadores da *Scene* 10% Imputada

Classificador	Algoritmo de Imputação
BR com J48	KNN
BR com NB	KNN
LP com J48	KNN
LP com NB	Mediana
RAKEL com J48	Mediana
RAKEL com NB	Média

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 52), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Scene* 10% Imputada, conforme pode ser visto no Gráfico 17.

Gráfico 17 - Melhor algoritmo de imputação para *Scene* 10% Imputada



Fonte: Autoria Própria

No Gráfico 17 é possível observar que o algoritmo Imputação KNN Iterativo é o melhor para a base *Scene* 10% Imputada, pois obteve 50% dos melhores resultados. A Imputação pela Mediana representa o segundo melhor algoritmo para a *Scene* 10% Imputada com 33,33% de valores mais próximos aos da base *Scene* Original. O terceiro melhor algoritmo de imputação foi o da Média correspondendo a 16,67% dos resultados mais próximos aos reais. Neste caso, o pior resultado foi do algoritmo Imputação pela Moda com 0% do total de melhores resultados.

4.2.5.2 *Scene* 20% Imputada

O Quadro 53 mostra os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Scene* com 20% de seus valores imputados.

Quadro 53 - Avaliação *Scene* 20% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,441	0,415	0,480	0,595	0,549	0,543
	Média	0,462	0,447	0,510	0,591	0,575	0,572
	Mediana	0,360	0,438	0,501	0,580	0,590	0,571
	KNN	0,523	0,465	0,569	0,605	0,599	0,592
HL	Moda	0,171	0,161	0,183	0,141	0,161	0,173
	Média	0,159	0,179	0,169	0,145	0,149	0,155
	Mediana	0,263	0,185	0,172	0,149	0,147	0,157
	KNN	0,141	0,180	0,152	0,140	0,150	0,146
Medida F	Moda	0,534	0,550	0,49	0,612	0,611	0,603
	Média	0,565	0,575	0,525	0,607	0,636	0,628
	Mediana	0,448	0,569	0,519	0,599	0,643	0,626
	KNN	0,611	0,586	0,574	0,623	0,647	0,647

Fonte: Autoria Própria

O Quadro 54 mostra a diferença das medidas de avaliação para a base *Scene* completa (Quadro 12) em relação a base *Scene* que teve 20% dos valores imputados (Quadro 53). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 54 - Diferença de avaliação *Scene* Original e 20% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,069	0,055	0,095	0,028	0,052	0,060
	Média	0,048	0,023	0,065	0,032	0,026	0,031
	Mediana	0,150	0,032	0,074	0,043	0,011	0,032
	KNN	-0,013	0,005	0,006	0,018	0,002	0,011
HL	Moda	-0,020	0,017	-0,034	-0,006	-0,015	-0,028
	Média	-0,008	-0,001	-0,020	-0,010	-0,003	-0,010
	Mediana	-0,112	-0,007	-0,023	-0,014	-0,001	-0,012
	KNN	0,010	-0,002	-0,003	-0,005	-0,004	-0,001
Medida F	Moda	0,062	0,039	0,097	0,027	0,037	0,051
	Média	0,031	0,014	0,062	0,032	0,012	0,026
	Mediana	0,148	0,020	0,068	0,040	0,005	0,028
	KNN	-0,015	0,003	0,013	0,016	0,001	0,007

Fonte: Autoria Própria

O Quadro 55 demonstra os melhores algoritmos de imputação para cada classificador aplicado na base *Scene* 20% Imputada, de acordo com o Quadro 54.

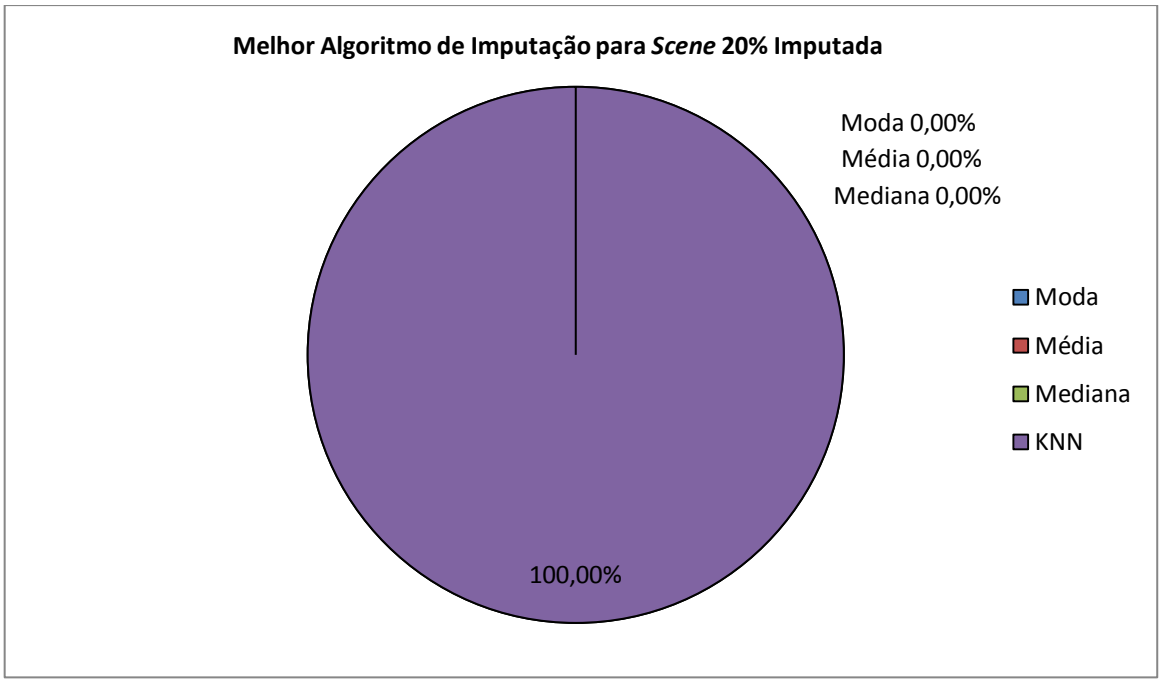
Quadro 55 - Melhores algoritmos de imputação para classificadores da *Scene* 20% Imputada

Classificador	Algoritmo de Imputação
BR com J48	KNN
BR com NB	KNN
LP com J48	KNN
LP com NB	KNN
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 55), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Scene* 20% Imputada, conforme é mostrado no Gráfico 18.

Gráfico 18 - Melhor algoritmo de imputação para *Scene* 20% Imputada



Fonte: Autoria Própria

O Gráfico 18 prova que o algoritmo Imputação KNN Iterativo foi o melhor para a base *Scene* 20% Imputada, pois totalizou 100% dos melhores resultados. Os

demais algoritmos de imputação não foram considerados como melhores para nenhum classificador aplicado, por isso correspondem a 0% do total.

4.2.5.3 *Scene* 30% Imputada

O Quadro 56 mostra os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Scene* com 30% de seus valores imputados.

Quadro 56 - Avaliação *Scene* 30% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,413	0,394	0,498	0,563	0,518	0,514
	Média	0,448	0,420	0,492	0,586	0,537	0,565
	Mediana	0,343	0,417	0,473	0,564	0,540	0,542
	KNN	0,493	0,449	0,530	0,605	0,584	0,575
HL	Moda	0,179	0,165	0,174	0,155	0,174	0,187
	Média	0,165	0,185	0,178	0,146	0,168	0,153
	Mediana	0,267	0,195	0,186	0,153	0,165	0,166
	KNN	0,160	0,186	0,167	0,139	0,149	0,152
Medida F	Moda	0,514	0,538	0,515	0,580	0,579	0,578
	Média	0,549	0,554	0,505	0,602	0,603	0,628
	Mediana	0,438	0,546	0,486	0,584	0,609	0,604
	KNN	0,573	0,571	0,537	0,624	0,635	0,633

Fonte: Autoria Própria

O Quadro 57 mostra a diferença das medidas de avaliação para a base *Scene* completa (Quadro 12) em relação a base *Scene* que teve 30% dos valores imputados (Quadro 56). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 57 - Diferença de avaliação *Scene* Original e 30% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,097	0,076	0,077	0,060	0,083	0,089
	Média	0,062	0,050	0,083	0,037	0,064	0,038
	Mediana	0,167	0,053	0,102	0,059	0,061	0,061
	KNN	0,017	0,021	0,045	0,018	0,017	0,028
HL	Moda	-0,028	0,013	-0,025	-0,020	-0,028	-0,042
	Média	-0,014	-0,007	-0,029	-0,011	-0,022	-0,008
	Mediana	-0,116	-0,017	-0,037	-0,018	-0,019	-0,021
	KNN	-0,009	-0,008	-0,018	-0,004	-0,003	-0,007
Medida F	Moda	0,082	0,051	0,072	0,059	0,069	0,076
	Média	0,047	0,035	0,082	0,037	0,045	0,026
	Mediana	0,158	0,043	0,101	0,055	0,039	0,050
	KNN	0,023	0,018	0,050	0,015	0,013	0,021

Fonte: Autoria Própria

O Quadro 58 se refere a base *Scene* 30% Imputada e mostra quais foram os melhores algoritmos de imputação para cada classificador, de acordo com o Quadro 57.

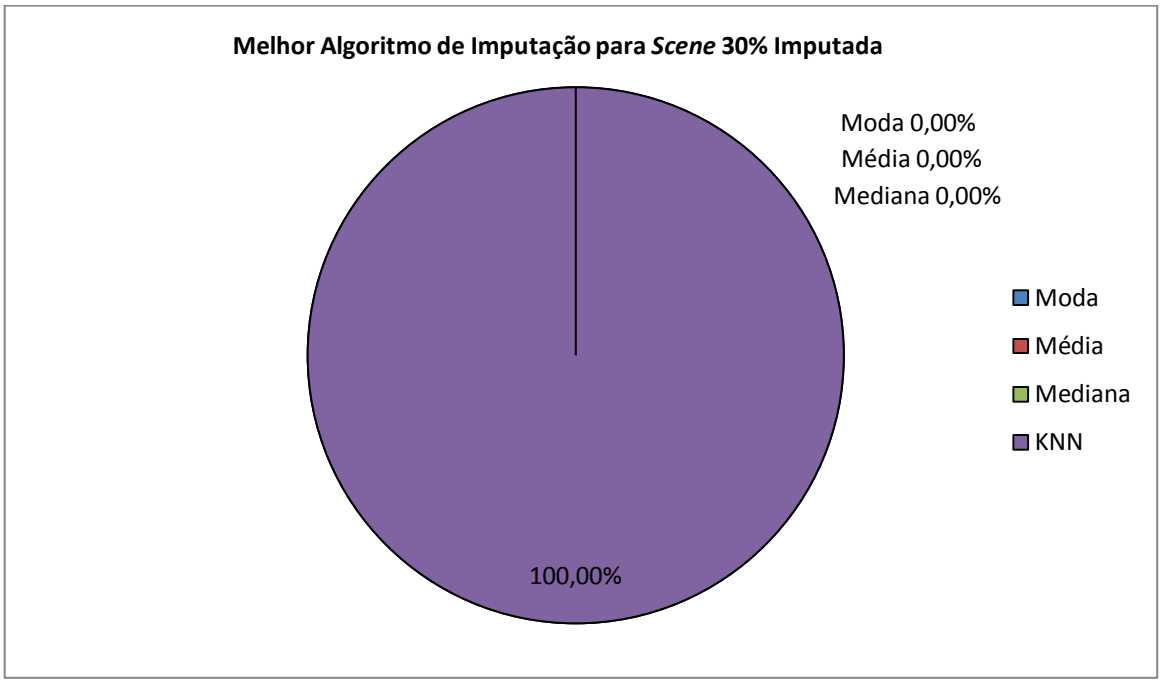
Quadro 58 - Melhores algoritmos de imputação para classificadores da *Scene* 30% Imputada

Classificador	Algoritmo de Imputação
BR com J48	KNN
BR com NB	KNN
LP com J48	KNN
LP com NB	KNN
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 58), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Scene* 30% Imputada, conforme pode ser visualizado no Gráfico 19.

Gráfico 19 - Melhor algoritmo de imputação para *Scene* 30% Imputada



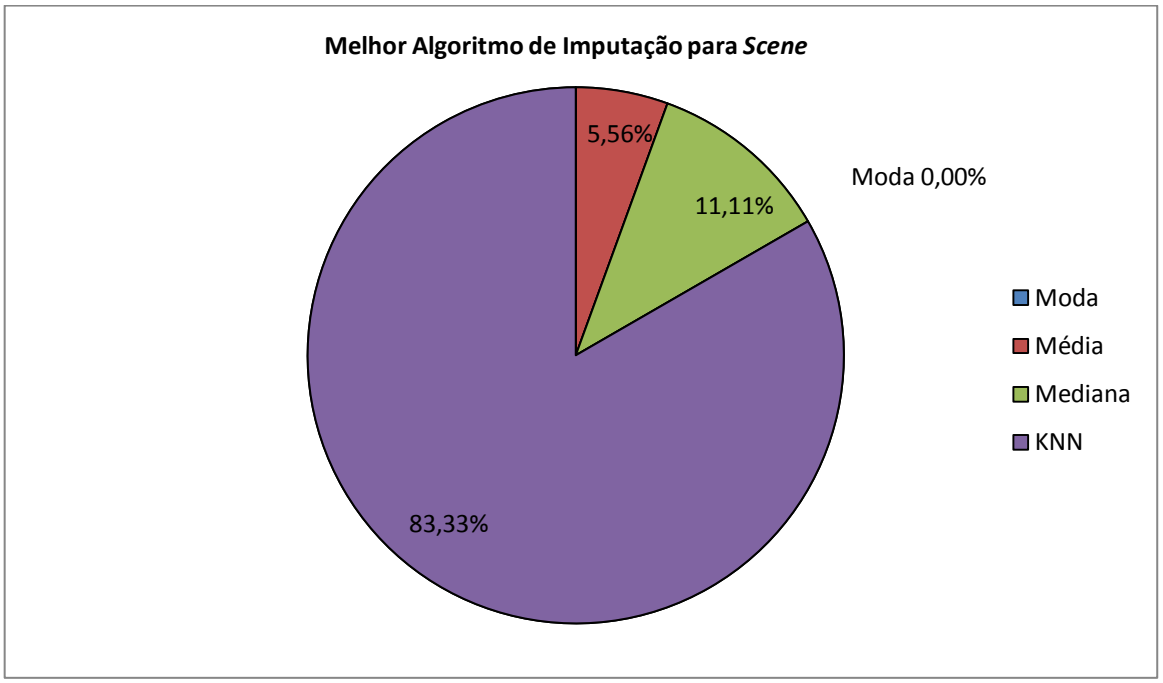
Fonte: Autoria Própria

O Gráfico 19 evidencia que o melhor algoritmo de imputação para a base *Scene* 30% Imputada é o KNN Iterativo, pois totalizou 100% dos melhores resultados. Os demais algoritmos de imputação não foram considerados como melhores para nenhum classificador aplicado, por isso correspondem a 0% do total.

4.2.5.4 Melhor algoritmo de imputação para *Scene*

A partir dos Gráficos 17, 18 e 19 pode-se calcular a média das porcentagens dos melhores algoritmos de imputação para as bases de dados *Scene* com 10%, 20% e 30% de valores imputados e obter o melhor algoritmo de imputação para as bases de dados *Scene*, conforme pode ser visualizado no Gráfico 20.

Gráfico 20 - Melhor algoritmo de imputação para *Scene*



Fonte: Autoria Própria

O Gráfico 20 demonstra que o algoritmo Imputação KNN Iterativo obteve 83,33% dos melhores resultados. Portanto, é considerado o melhor algoritmo de imputação para as bases *Scene*. Imputação pela Mediana é o segundo melhor algoritmo para as bases *Scene* que foram imputadas representando 11,11% dos melhores resultados. O algoritmo Imputação pela Média foi o terceiro melhor para as bases de dados *Scene*. Imputação pela Moda apresentou os piores resultados, no geral corresponde a 0% dos resultados que mais se aproximaram dos valores reais.

4.2.6 Base de dados *Yeast*

Foram realizados experimentos com três porcentagens diferentes de valores imputados na base de dados *Yeast*, por isso os resultados serão apresentados separadamente para cada porcentagem.

4.2.6.1 *Yeast* 10% Imputada

O Quadro 59 apresenta os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os

classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Yeast* com 10% de seus valores imputados.

Quadro 59 - Avaliação *Yeast* 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,297	0,451	0,367	0,475	0,409	0,426
	Média	0,325	0,452	0,386	0,467	0,412	0,428
	Mediana	0,317	0,452	0,368	0,469	0,410	0,426
	KNN	0,326	0,453	0,382	0,467	0,411	0,425
HL	Moda	0,506	0,252	0,300	0,238	0,326	0,286
	Média	0,443	0,252	0,292	0,242	0,327	0,288
	Mediana	0,477	0,252	0,299	0,240	0,324	0,288
	KNN	0,430	0,251	0,294	0,242	0,329	0,290
Medida F	Moda	0,454	0,583	0,499	0,598	0,559	0,562
	Média	0,480	0,583	0,516	0,592	0,563	0,561
	Mediana	0,474	0,583	0,497	0,595	0,560	0,561
	KNN	0,482	0,584	0,510	0,593	0,557	0,560

Fonte: Autoria Própria

O Quadro 60 mostra a diferença das medidas de avaliação para a base *Yeast* completa (Quadro 13) em relação a base *Yeast* 10% Imputada (Quadro 59).

Quadro 60 - Diferença de avaliação *Yeast* Original e 10% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	-0,002	-0,001	0,025	-0,019	0,015	-0,003
	Média	-0,030	-0,002	0,006	-0,011	0,012	-0,005
	Mediana	-0,022	-0,002	0,024	-0,013	0,014	-0,003
	KNN	-0,031	-0,003	0,010	-0,011	0,013	-0,002
HL	Moda	-0,011	0,001	-0,017	0,008	-0,018	0,005
	Média	0,052	0,001	-0,009	0,004	-0,019	0,003
	Mediana	0,018	0,001	-0,016	0,006	-0,016	0,003
	KNN	0,065	0,002	-0,011	0,004	-0,021	0,001
Medida F	Moda	-0,002	-0,002	0,029	-0,015	0,008	-0,003
	Média	-0,028	-0,002	0,012	-0,009	0,004	-0,002
	Mediana	-0,022	-0,002	0,031	-0,012	0,007	-0,002
	KNN	-0,030	-0,003	0,018	-0,010	0,010	-0,001

Fonte: Autoria Própria

O Quadro 61 mostra os melhores algoritmos de imputação para cada classificador aplicado na base *Yeast* 10% Imputada, de acordo com o Quadro 60.

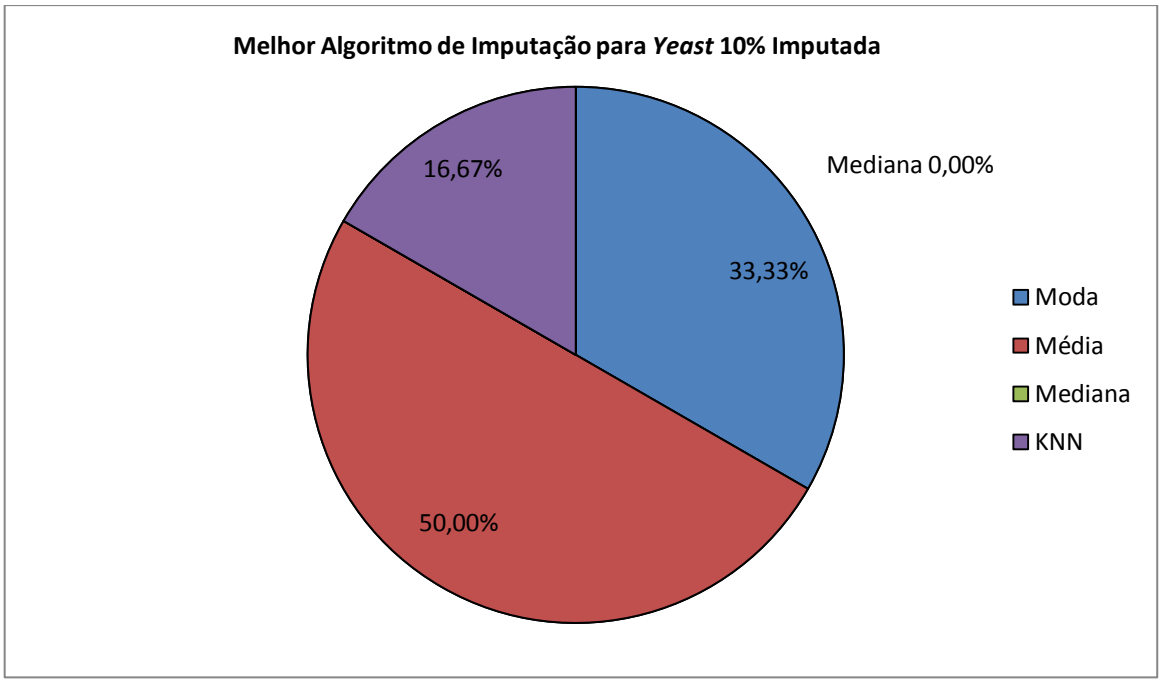
Quadro 61 - Melhores algoritmos de imputação para classificadores da *Yeast* 10% Imputada

Classificador	Algoritmo de Imputação
BR com J48	Moda
BR com NB	Moda
LP com J48	Média
LP com NB	Média
RAKEL com J48	Média
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 61), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Yeast* 10% Imputada, conforme é mostrado no Gráfico 21.

Gráfico 21 - Melhor algoritmo de imputação para *Yeast* 10% Imputada



Fonte: Autoria Própria

No Gráfico 21 é possível observar que o melhor algoritmo de imputação para a base *Yeast* 10% Imputada foi o que utilizou a Média, pois obteve 50% dos melhores resultados. O segundo melhor algoritmo foi a Imputação pela Moda com

33,33% dos resultados mais próximos aos reais. Em seguida, aparece o algoritmo KNN Iterativo que corresponde a 16,67% dos melhores resultados. Neste caso, o pior algoritmo de imputação foi o da Mediana que representa 0% dos resultados que mais se aproximaram aos da *Yeast* Original.

4.2.6.2 *Yeast* 20% Imputada

O Quadro 62 mostra os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Yeast* com 20% de seus valores imputados.

Quadro 62 - Avaliação *Yeast* 20% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,284	0,443	0,384	0,454	0,401	0,421
	Média	0,309	0,440	0,392	0,460	0,406	0,422
	Mediana	0,291	0,442	0,379	0,460	0,400	0,419
	KNN	0,333	0,443	0,403	0,465	0,406	0,419
HL	Moda	0,479	0,257	0,301	0,247	0,333	0,288
	Média	0,461	0,257	0,289	0,244	0,331	0,283
	Mediana	0,491	0,257	0,297	0,245	0,336	0,289
	KNN	0,430	0,255	0,281	0,243	0,328	0,289
Medida F	Moda	0,437	0,575	0,506	0,581	0,549	0,555
	Média	0,464	0,574	0,519	0,587	0,552	0,557
	Mediana	0,445	0,575	0,507	0,588	0,545	0,553
	KNN	0,489	0,578	0,533	0,592	0,552	0,553

Fonte: Autoria Própria

O Quadro 63 mostra a diferença das medidas de avaliação para a base *Yeast* completa (Quadro 13) em relação a base *Yeast* que teve 20% dos valores imputados (Quadro 62). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 63 - Diferença de avaliação *Yeast* Original e 20% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,011	0,007	0,008	0,002	0,023	0,002
	Média	-0,014	0,010	0,000	-0,004	0,018	0,001
	Mediana	0,004	0,008	0,013	-0,004	0,024	0,004
	KNN	-0,038	0,007	-0,011	-0,009	0,018	0,004
HL	Moda	0,016	-0,004	-0,018	-0,001	-0,025	0,003
	Média	0,034	-0,004	-0,006	0,002	-0,023	0,008
	Mediana	0,004	-0,004	-0,014	0,001	-0,028	0,002
	KNN	0,065	-0,002	0,002	0,003	-0,020	0,002
Medida F	Moda	0,015	0,006	0,022	0,002	0,018	0,004
	Média	-0,012	0,007	0,009	-0,004	0,015	0,002
	Mediana	0,007	0,006	0,021	-0,005	0,022	0,006
	KNN	-0,037	0,003	-0,005	-0,009	0,015	0,006

Fonte: Autoria Própria

O Quadro 64 é referente a base *Yeast* 20% Imputada e mostra quais foram os melhores algoritmos de imputação para cada classificador, de acordo com o Quadro 63.

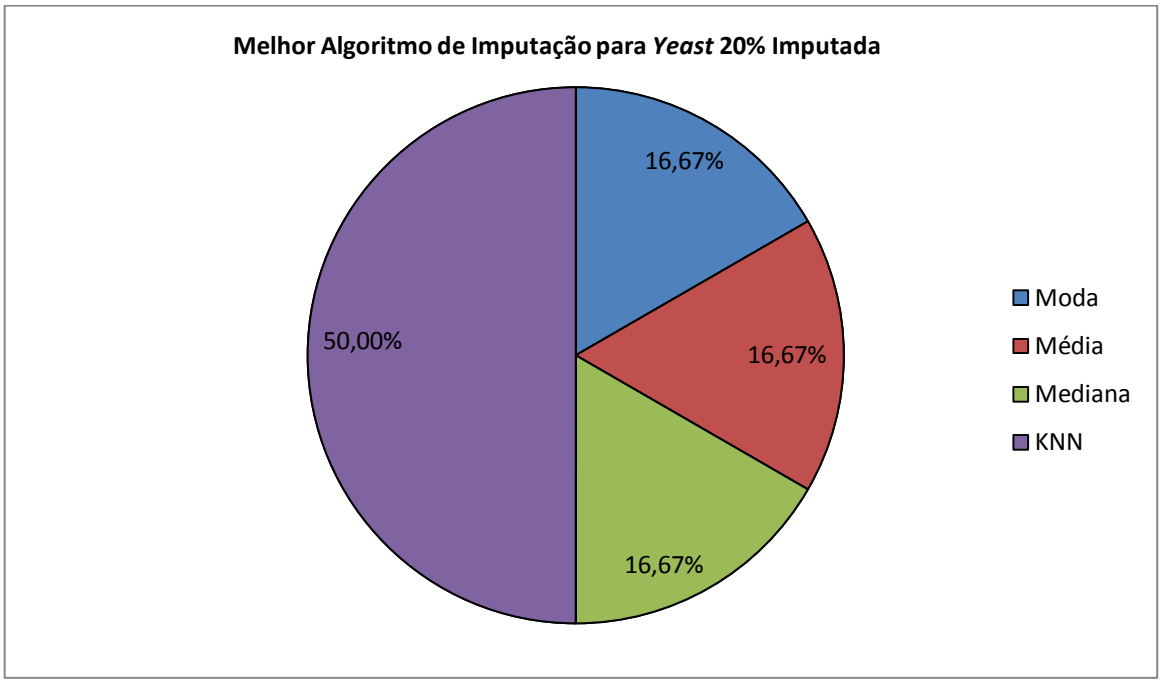
Quadro 64 - Melhores algoritmos de imputação para classificadores da *Yeast* 20% Imputada

Classificador	Algoritmo de Imputação
BR com J48	Mediana
BR com NB	KNN
LP com J48	KNN
LP com NB	Moda
RAKEL com J48	KNN
RAKEL com NB	Média

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 64), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Yeast* 20% Imputada, conforme pode ser visualizado no Gráfico 22.

Gráfico 22 - Melhor algoritmo de imputação para *Yeast* 20% Imputada



Fonte: Autoria Própria

O Gráfico 22 mostra que o melhor algoritmo de Imputação para a base *Yeast* 20% Imputada foi o KNN Iterativo, pois obteve 50% dos melhores resultados. Os demais algoritmos de imputação tiveram percentuais iguais de melhores resultados, por isso qualquer um é indicado como melhor algoritmo de imputação depois do KNN Iterativo.

4.2.6.3 *Yeast* 30% Imputada

O Quadro 65 apresenta os valores das medidas de avaliação Acurácia, HL e Medida F para os classificadores multirrótulo BR, LP e RAKEL combinados com os classificadores monorrótulo J48 e NB que foram aplicados nas bases de dados *Yeast* com 30% de seus valores imputados.

Quadro 65 - Avaliação *Yeast* 30% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,289	0,431	0,362	0,453	0,391	0,414
	Média	0,306	0,435	0,365	0,441	0,394	0,412
	Mediana	0,299	0,436	0,373	0,446	0,399	0,407
	KNN	0,336	0,438	0,381	0,461	0,412	0,420
HL	Moda	0,487	0,262	0,303	0,248	0,348	0,300
	Média	0,449	0,259	0,302	0,256	0,340	0,299
	Mediana	0,442	0,259	0,294	0,254	0,338	0,304
	KNN	0,383	0,259	0,293	0,245	0,324	0,293
Medida F	Moda	0,443	0,565	0,492	0,581	0,537	0,549
	Média	0,465	0,571	0,497	0,568	0,544	0,548
	Mediana	0,455	0,571	0,508	0,572	0,548	0,543
	KNN	0,491	0,571	0,512	0,590	0,563	0,555

Fonte: Autoria Própria

O Quadro 66 mostra a diferença das medidas de avaliação para a base *Yeast* completa (Quadro 13) em relação a base *Yeast* que teve 30% dos valores imputados (Quadro 65). Os valores das medidas de avaliação que mais se aproximaram dos valores reais estão sublinhados.

Quadro 66 - Diferença de avaliação *Yeast* Original e 30% Imputada

	Algoritmos de Imputação	BR		LP		RAKEL	
		J48	NB	J48	NB	J48	NB
Acurácia	Moda	0,006	0,019	0,030	0,003	0,033	0,009
	Média	-0,011	0,015	0,027	0,015	0,030	0,011
	Mediana	-0,004	0,014	0,019	0,010	0,025	0,016
	KNN	-0,041	0,012	0,011	-0,005	0,012	0,003
HL	Moda	0,008	-0,009	-0,020	-0,002	-0,040	-0,009
	Média	0,046	-0,006	-0,019	-0,010	-0,032	-0,008
	Mediana	0,053	-0,006	-0,011	-0,008	-0,030	-0,013
	KNN	0,112	-0,006	-0,010	0,001	-0,016	-0,002
Medida F	Moda	0,009	0,016	0,036	0,002	0,030	0,010
	Média	-0,013	0,010	0,031	0,015	0,023	0,011
	Mediana	-0,003	0,010	0,020	0,011	0,019	0,016
	KNN	-0,039	0,010	0,016	-0,007	0,004	0,004

Fonte: Autoria Própria

O Quadro 67 mostra os melhores algoritmos de imputação para cada classificador aplicado na base *Yeast* 30% Imputada, de acordo com o Quadro 66.

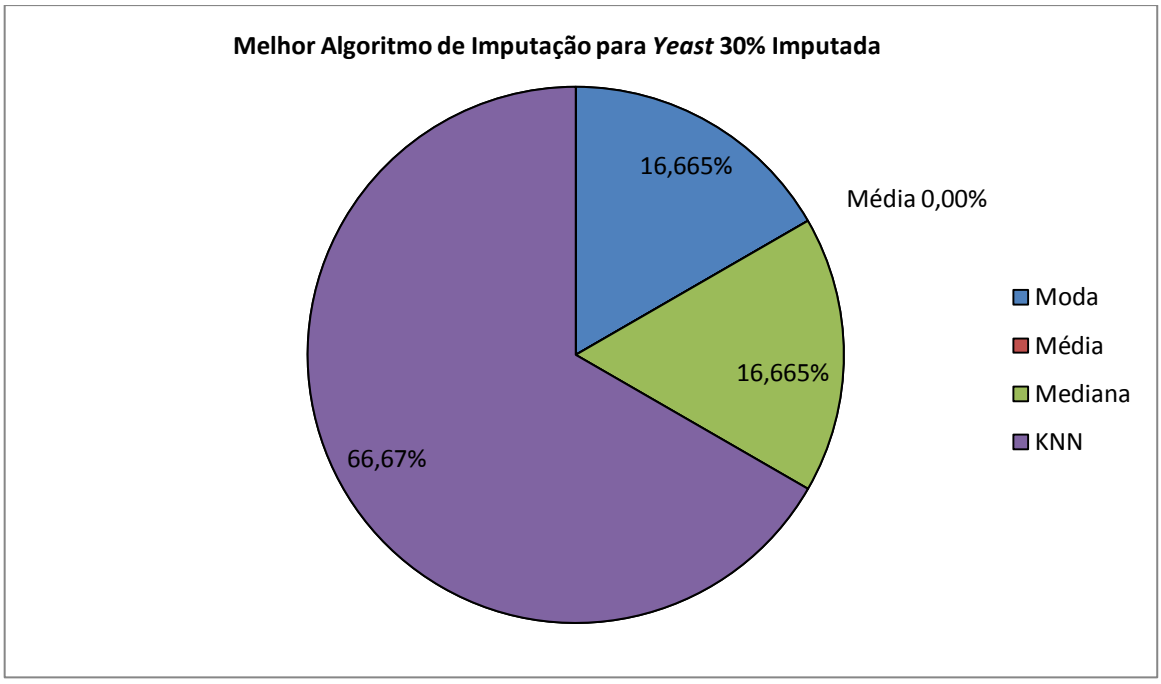
Quadro 67 - Melhores algoritmos de imputação para classificadores da *Yeast* 30% Imputada

Classificador	Algoritmo de Imputação
BR com J48	Mediana
BR com NB	KNN
LP com J48	KNN
LP com NB	Moda
RAKEL com J48	KNN
RAKEL com NB	KNN

Fonte: Autoria Própria

A partir dos melhores algoritmos de imputação encontrados para cada classificador (Quadro 67), pode-se calcular as porcentagens que cada um representa como melhor algoritmo de imputação para a base de dados *Yeast* 30% Imputada, conforme é mostrado no Gráfico 23.

Gráfico 23 - Melhor algoritmo de imputação para *Yeast* 30% Imputada



Fonte: Autoria Própria

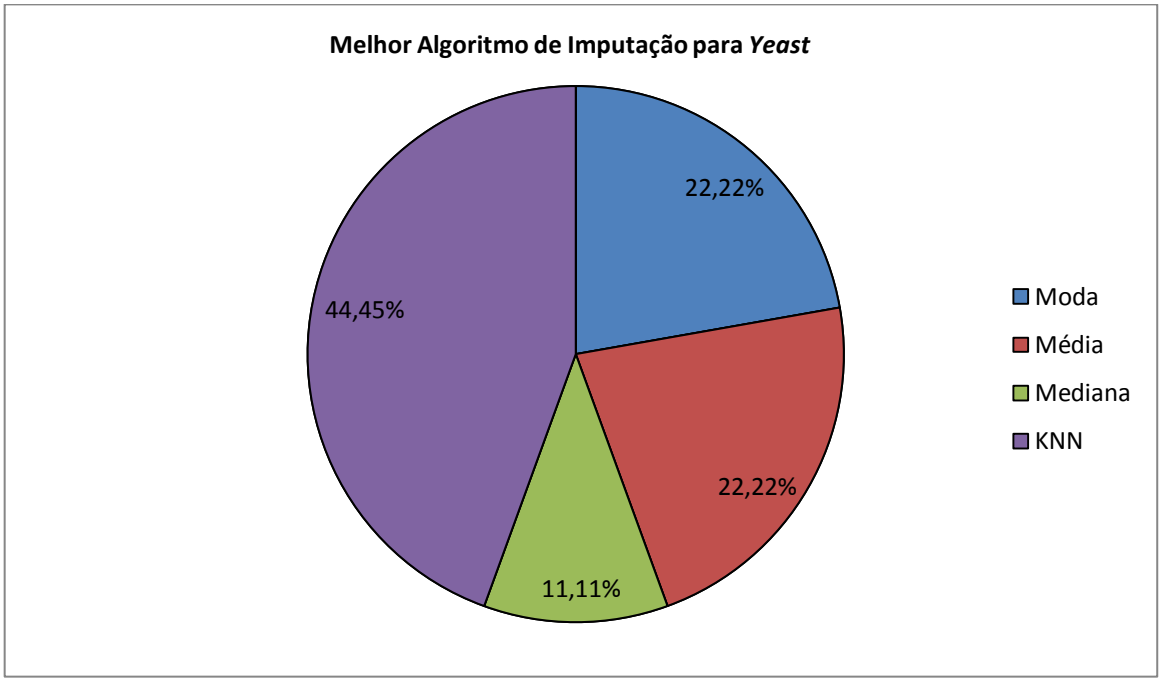
No Gráfico 23 é possível observar que o melhor algoritmo de Imputação para a base *Yeast* 30% Imputada foi o KNN Iterativo, pois obteve 66,67% dos resultados mais próximos aos da base *Yeast* Original. Os algoritmos Imputação pela Moda e

Imputação pela Mediana tiveram percentuais iguais de melhores resultados, por isso são indicados para a base *Yeast* 30% Imputada após o algoritmo Imputação KNN Iterativo. O pior resultado ficou para o algoritmo Imputação pela Média que corresponde a 0% do total.

4.2.6.4 Melhor algoritmo de imputação para *Yeast*

A partir dos Gráficos 21, 22 e 23 pode-se calcular a média das porcentagens dos melhores algoritmos de imputação para as bases de dados *Yeast* com 10%, 20% e 30% de valores imputados e obter o melhor algoritmo de imputação para as bases de dados *Yeast*, conforme é apresentado no Gráfico 24.

Gráfico 24 - Melhor algoritmo de imputação para *Yeast*



Fonte: Autoria Própria

O Gráfico 24 demonstra que o algoritmo Imputação KNN Iterativo é o melhor para as bases *Scene* representando 44,45% dos melhores resultados. Imputação pela Moda e Imputação pela Média podem ser considerados como o segundo melhor algoritmo de imputação para as bases *Scene* que foram imputadas, pois cada um representa 22,22% dos resultados mais próximos aos reais. O algoritmo Imputação pela Mediana teve o pior resultado correspondendo a 11,11% dos resultados que mais se aproximaram da base de dados original.

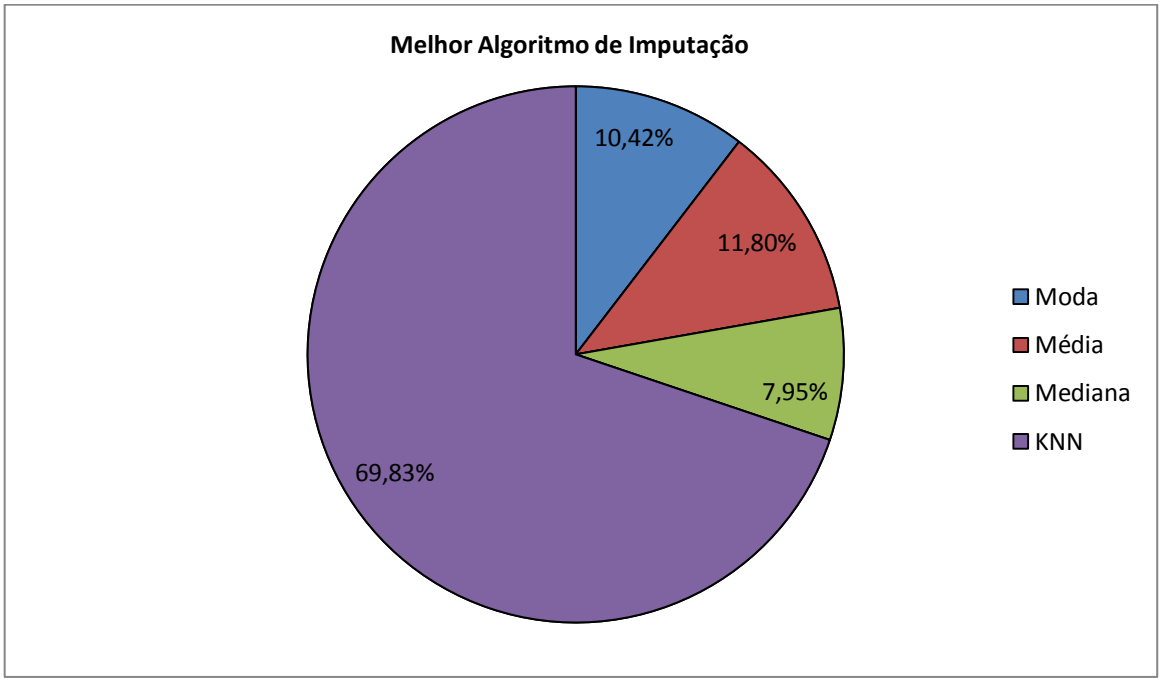
4.3 MELHOR ALGORITMO DE IMPUTAÇÃO

Na seção 4.2 foram analisados os resultados das bases de dados que passaram pelo processo de imputação com diferentes porcentagens de valores faltantes e, em seguida, foram submetidas a classificadores. A partir dos resultados das medidas Acurácia, HL e Medida F que avaliaram os classificadores aplicados nas bases imputadas, foram identificados os algoritmos de imputação que alcançaram valores mais próximos aos das bases de dados originais. Por fim, foram encontrados os melhores algoritmos de imputação para cada base de dados.

Esta seção tem como objetivo mostrar o melhor algoritmo de imputação no geral, ou seja, o algoritmo que obteve os melhores resultados considerando todas as bases de dados utilizadas neste trabalho.

O melhor algoritmo de imputação pode ser encontrado através do cálculo da média das porcentagens dos melhores algoritmos de imputação de todas as bases de dados (Gráficos 4, 8, 12, 16, 20 e 24), conforme pode ser visualizado no Gráfico 25.

Gráfico 25 - Melhor algoritmo de imputação



Fonte: Autoria Própria

O Gráfico 25 prova que o algoritmo Imputação KNN Iterativo foi o melhor entre os algoritmos de imputação testados. Da mesma forma que ocorreu na

avaliação dos algoritmos de imputação para cada base de dados, o KNN Iterativo representa a maior parte dos melhores resultados, correspondendo a 69,83%. Os demais algoritmos de imputação representam juntos 30,17% dos melhores resultados, ou seja, juntos não chegam nem a metade da porcentagem alcançada pelo KNN Iterativo. De modo geral, o algoritmo Imputação pela Média foi o segundo melhor para as bases utilizadas, correspondendo a 11,80% dos resultados mais próximos aos reais. Em terceiro lugar, aparece o algoritmo Imputação pela Moda equivalendo a 10,42% do total. O pior resultado foi do algoritmo Imputação pela Mediana que representa apenas 7,95% dos melhores resultados.

No geral, o algoritmo imputação KNN Iterativo alcançou os resultados mais próximos aos das bases de dados originais. Este algoritmo se destacou em relação aos demais algoritmos de imputação devido ao fato de ele ser mais elaborado e utilizar aprendizagem de máquina. Os algoritmos Imputação pela Moda, Imputação pela Média e Imputação pela Mediana são mais simples e não possuem nenhum tipo de inteligência implementada, isso explica porque eles tiveram resultados piores que o KNN Iterativo.

O algoritmo Imputação pela Mediana teve os piores resultados, representando 7,95% dos resultados que mais se aproximaram dos valores reais. Uma explicação para isso é que a Mediana não é uma boa medida para valores que fogem da tendência central de um conjunto. Em outras palavras, os valores das Medianas não são foram representativos para os conjuntos de dados utilizados.

Neste trabalho, foram analisados os resultados de 18 bases de dados imputadas e cada base teve um melhor algoritmo de imputação. O melhor algoritmo de imputação para cada base foi aquele que alcançou a maioria dos melhores resultados, ou seja, dos resultados mais próximos aos reais. A Tabela 5 mostra a frequência, ou seja, o número de casos em que cada algoritmo de imputação foi considerado o melhor entre os 18 casos possíveis.

Tabela 5 - Frequência melhor algoritmo de imputação

Algoritmo de Imputação	Nº de casos de melhor algoritmo de imputação
Imputação pela Moda	0
Imputação pela Média	1
Imputação pela Mediana	0
Imputação KNN Iterativo	17

Fonte: Autoria própria

Como é possível observar na Tabela 5, o algoritmo KNN Iterativo foi o melhor algoritmo de imputação para 17 entre 18 casos testados. Por isso, ele é o algoritmo mais indicado para ser aplicado em bases de dados multirrótulo com valores faltantes que irão passar pelo processo de classificação multirrótulo. Além de apresentar a maior porcentagem de melhores resultados (Gráfico 25), o KNN Iterativo foi considerado o melhor algoritmo de imputação para praticamente todas as bases de dados.

De acordo com a Tabela 5, houve um caso em que o KNN não foi considerado o melhor algoritmo de imputação. Na base *Yeast* 10% Imputada o melhor algoritmo foi Imputação pela Média, pois foi o que melhor se ajustou aos dados contidos nesta base para substituir os valores que estavam faltando. O algoritmo Imputação pela Média juntamente com os algoritmos Imputação pela Moda e Mediana apresentaram resultados inferiores ao KNN Iterativo.

Os resultados obtidos neste trabalho também foram alcançados no trabalho de Acuña e Rodriguez (2014), onde foram realizados testes com os algoritmos de imputação pela Média, Mediana e KNN, porém utilizando bases de dados monorrótulo. O KNN apresentou os melhores resultados em relação aos demais algoritmos.

Os resultados também se repetiram no trabalho de Batista e Monard (2003), onde foram testados os algoritmos de imputação pela Média, Moda e KNN em bases de dados monorrótulo. Novamente, o KNN apresentou os melhores resultados em relação aos outros algoritmos de imputação.

Os resultados alcançados neste trabalho, juntamente com os resultados dos trabalhos de Acuña e Rodriguez (2014) e Batista e Monard (2003), comprovam que o algoritmo Imputação KNN Iterativo é melhor que os algoritmos Imputação pela

Moda, Média e Mediana e pode ser aplicado tanto em bases de dados monorrótulo, como em bases de dados multirrótulo.

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi realizada uma análise da aplicação de algoritmos de imputação de valores em bases de dados multirrótulo para a utilização da tarefa de classificação. Foram testados os algoritmos Imputação pela Moda, Média, Mediana e KNN Iterativo com o objetivo de encontrar os que possuem melhores resultados, ou seja, que alcançam valores mais próximos aos de bases de dados completas e originais.

Esta análise foi motivada devido ao problema de valores faltantes em bases de dados que podem limitar o uso da classificação multirrótulo em diversos domínios de aplicação. Para que a classificação multirrótulo possa ocorrer é necessário que ocorra o pré-processamento dos dados omissos, ou seja, deve ocorrer o processo de imputação para que valores incompletos sejam substituídos por valores estimados mais próximos aos reais.

Os algoritmos de imputação foram investigados para seis bases de dados multirrótulo com diferentes porcentagens de valores faltantes, sendo 10%, 20% e 30%. Cada base de dados imputada passou pelos classificadores multirrótulo: BR, LP e RAKEL. Aplicados com os classificadores monorrótulo: J48 e *Naive Bayes*. As medidas de avaliação Acurácia, HL e Medida F avaliaram os resultados. Os resultados das bases de dados imputadas foram comparados com os das bases de dados originais a fim de verificar os algoritmos de imputação que alcançaram os melhores resultados.

De modo geral, o algoritmo KNN Iterativo foi superior aos algoritmos Imputação pela Moda, Média e Mediana devido ao fato de utilizar aprendizagem de máquina e os outros não. O KNN Iterativo obteve a maioria dos melhores resultados, representando 69,83% do total. O pior resultado foi do algoritmo Imputação pela Mediana, correspondendo a apenas 7,95% dos melhores resultados. Os algoritmos Imputação pela Média e Imputação pela Moda alcançaram resultados melhores que do algoritmo Imputação pela Mediana, porém não se distanciaram muito do pior colocado.

O algoritmo KNN Iterativo foi o melhor algoritmo de imputação para 17 entre 18 casos testados, representando 94,44% do total de casos possíveis para ser o melhor algoritmo de imputação. Por isso, se houver um caso em que se deve

escolher um dos quatro algoritmos testados é aconselhável optar pelo KNN Iterativo ao invés de Imputação pela Moda, Média e Mediana. Os resultados obtidos comprovam a superioridade do KNN Iterativo diante dos demais algoritmos de imputação.

É importante ressaltar que os valores estimados nas imputações são de boa qualidade, já que havia o conhecimento dos valores reais dos dados contidos nas bases originais, das quais foram retirados valores por meio de sorteios aleatórios a fim de simular bases de dados com valores faltantes para passarem pelo processo de classificação e, então, terem seus resultados comparados com os valores reais.

Com este trabalho, espera-se que tenha sido evidenciada a importância de analisar algoritmos de imputação que serão aplicados em bases de dados multirrótulo para a tarefa de classificação, uma vez que algoritmos mal escolhidos podem trazer resultados ruins e algoritmos selecionados adequadamente podem resultar em valores iguais ou bem próximos aos valores reais.

5.1 TRABALHOS FUTUROS

Para trabalhos futuros seria interessante aplicar os algoritmos Imputação pela Moda, Média, Mediana e KNN Iterativo em outras bases de dados multirrótulo a fim de complementar a veracidade de que o KNN Iterativo é o melhor entre os algoritmos de imputação testados. Também seria válido aplicar as bases de dados imputadas em outros classificadores para verificar se os resultados dos algoritmos de imputação continuariam os mesmos.

Outra possibilidade seria modificar o valor de *k* do algoritmo Imputação KNN Iterativo para aumentar sua eficiência. Além disso, poderia ser alterado o passo inicial do KNN Iterativo, ao invés de realizar uma Imputação pela Média, poderia ser aplicada Imputação pela Moda ou Mediana para completar a base de dados com valores faltantes e, então, continuar a execução do algoritmo normalmente.

Por fim, poderiam ser aplicados algoritmos de imputação que utilizam aprendizagem de máquina, como Imputação utilizando *k-Means*, nas mesmas ou em outras bases de dados multirrótulo com o objetivo de verificar se o KNN Iterativo se

destacaria por alcançar os melhores resultados da mesma maneira que ocorreu neste trabalho.

REFERÊNCIAS

ACUÑA, Edgar; RODRIGUEZ, Caroline. The treatment of missing values and its effect in the classifier accuracy. **Classification, Clustering and Data Mining Aplications**, Chicago, p. 639-647, jul. 2004.

BATISTA, Gustavo E. A. P. A. **Pré-processamento de dados em aprendizado de máquina supervisionado**. 2003. 232 f. Tese (Doutorado em Ciências - Ciências de Computação e Matemática Computacional) - Universidade de São Paulo, São Carlos, 2003.

BATISTA, Gustavo E. A. P. A.; MONARD, Maria C. An analysis of four missing data treatment methods for supervised learning, 2003. Disponível em: <http://conteudo.icmc.usp.br/pessoas/gbatista/files/aai2003.pdf>. Acesso em: 28 nov. 2016.

BEZERRA, Miguel E. R. **Métodos baseados na regra do vizinho mais próximo para reconhecimento de imagens**. 2006. 90 f. Trabalho de Conclusão de Curso - Curso Superior de Engenharia da Computação. Escola Politécnica de Pernambuco - Universidade de Pernambuco, Recife, 2006.

BORGES, Helyane B. **Classificador hierárquico multirrótulo usando uma rede neural competitiva**. 2012. 188 f. Tese (Doutorado em Informática) - Universidade Católica do Paraná, Curitiba, 2012.

BRITTO, Usiara. **Imputação de dados utilizando o algoritmo EM e regressão linear no SPSS**. 2005. 66 f. Trabalho de Conclusão de Curso - Curso Superior de Estatística. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005.

CASTRO, Isabela Q. **Uma aplicação de métodos de imputação no estudo de fatores associados ao baixo peso ao nascer**. 2014. 78 f. Trabalho de Conclusão de Curso - Curso Superior de Estatística. Universidade Federal de Juiz de Fora, Juiz de Fora, 2014.

CERRI, Ricardo. **Técnicas de classificação hierárquica multirrótulo**. 2010. 241 f. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) - Universidade de São Paulo, São Carlos, 2010.

CHERMAN, Everton A.; METZ, Jean; MONARD, Maria C. Explorando dependências entre rótulos no classificador multirrótulo Binary Relevance. In: WORKSHOP ON

COMPUTATIONAL INTELLIGENCE, 3, 2010, São Bernardo do Campo. **Anais…**São Bernardo do Campo: WCI, 2010.

COELHO, Tiago A.; ESMIN, Ahmed A. A.; JÚNIOR, Wagner M. Uma estratégia híbrida para o problema de classificação multirrótulo. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL, 8, 2011, Natal. **Anais...** Natal: ENIA, 2011.

GAMA, Patrícia P.; BERNARDINI, Flavia C.; ZADROZNY, Bianca. Proposta de um novo método para classificação multirrótulo baseado em seleção aleatória e *bagging.* In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL, 9, 2012, Curitiba. **Anais...** Curitiba: ENIA, 2012.

GHINOZZI, Glauder G. **Aprendizado semissupervisionado aplicado ao problema de valores ausentes**. 2012. 96 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Mato Grosso do Sul, Campo Grande, 2012.

GIASSON, Elvio *et al*. Avaliação de cinco algoritmos de árvores de decisão e três tipos de modelos digitais de elevação para mapeamento digital de solos a nível semidetalhado na Bacia do Lageado Grande, RS, Brasil. **Ciência Rural**, Santa Maria, v. 43, n. 11, nov. 2013. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-84782013001100008>. Acesso em: 16 mai. 2017.

METZ, Jean. **Abordagens para aprendizado semissupervisionado multirrótulo e hierárquico**. 2011. 219 f. Tese (Doutorado em Ciências - Ciências de Computação e Matemática Computacional) - Universidade de São Paulo, São Carlos, 2011.

MITCHELL, Tom M. **Machine Learning**. McGraw-Hill Higher Education, 1997.

MUNDFROM, Daniel J.; WHITCOMB, Alan. Imputing missing values: the effect on the accuracy of classification. **Multiple Regression Viewpoints**, v. 25, p. 13-19, 1998. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.523.5167&rep=rep1&type=pdf>. Acesso em: 28 nov. 2016.

NUNES, Luciana N. **Métodos de Imputação de Dados Aplicados na Área da Saúde**. 2007. 120 f. Tese (Doutorado em Epidemiologia) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007.

NUNES, Luciana N.; KLÜCK, Mariza M.; FACHEL, Jandyra M. G**.** Uso da imputação múltipla de dados faltantes: uma simulação utilizando dados epidemiológicos. **Caderno de Saúde Pública**, Rio de Janeiro, v. 25, n. 2, p. 268-278, fev. 2009.

OLIVEIRA, Pedro G. **Imputação automática de atributos faltantes em problemas de classificação:** um estudo comparativo envolvendo algoritmos bio-inspirados. 2009. 98 f. Dissertação (Mestrado em Informática Aplicada) - Universidade de Fortaleza, Fortaleza, 2009.

OLIVEIRA, Werbeston D. **Comparação dos algoritmos C4.5 e MLP usados na avaliação da segurança dinâmica e no auxílio ao controle preventivo no contexto da estabilidade transitória de sistemas de potência**. 2013. 110 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Pará, Belém, 2013.

OLSON, David L.; DELEN, Dursun. **Advanced Data Mining Techniques**. Berlin: Springer, 2008.

PATIL, Tina R.; Sherekar, S. Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classication. International Journal of Computer Science and Applications, abr. 2013. Disponível em: <http://researchpublications.org/IJCSA/NCAICN-13/189.pdf>. Acesso em: 16 mai. 2017.

PIMENTEL, Edson P.; FRANÇA, Vilma F.; OMAR, Nizam. A identificação de grupos de aprendizes no ensino presencial utilizando técnicas de clusterização. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 14, 2003, Rio de Janeiro. **Anais**... Rio de Janeiro: SBIE, 2003.

PRATI, Ronaldo C. Uso de predição conformal em classificação multirrótulo: resultados iniciais. In: SYMPOSIUM ON KNOWLEDGE DISCOVERY, MINING AND LEARNING, 2013, São Carlos. **Anais...** São Carlos: KDMile, 2013.

READ, Jesse *et al*. Meka: A Multi-label / Multi-target Extension to Weka. Journal of Machine Learning Research 17, fev. 2016. Disponível em: < http://www.jmlr.org/papers/v17/12-164.html>. Acesso em: 16 mai. 2017.

REZENDE, S, O. **Sistemas inteligentes:** Fundamentos e Aplicações. 1. ed. Barueri: Manole, 2005.

RIVOLLI, Adriano; CARVALHO, André C. P. L. C. O uso seletivo de classificadores binários na solução de problemas multirrótulos. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL, 12, 2015, Natal. **Anais...** Natal: ENIA, 2015.

RODRIGUES, Filipe M. **Uso de confiabilidade na rotulação de exemplos em problemas de classificação multirrótulo com aprendizado semissupervisionado**. 2014. 118 f. Dissertação (Mestrado em Sistemas e Computação) - Universidade Federal do Rio Grande do Norte, Natal, 2014.

SANTOS, Araken M. **Investigando a combinação de técnicas de aprendizado semissupervisionado e classificação hierárquica multirrótulo**. 2012. 214 f. Tese (Doutorado em Sistemas e Computação) - Universidade Federal do Rio Grande do Norte, Natal, 2012.

SILVA, Alexandre B. M. **Redes neurais artificiais, análise de sensibilidade e o comportamento de funções de comércio exterior do Brasil**. 2002. 219 f. Tese (Doutorado em Economia) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

SILVA, Marcelo M. **Uma abordagem evolucionária para o aprendizado semi-supervisionado em máquinas de vetores de suporte**. 2008. 106 f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Minas Gerais, Belo Horizonte, 2008.

SILVA, Jonathan A. **Substituição de valores ausentes: uma abordagem baseada em um algoritmo evolutivo para agrupamento de dados**. 2010. 150 f. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) – Universidade de São Paulo, São Carlos, 2010.

SILVA, Antonio G. **Uma análise da aplicação dos métodos de classificação multirrótulo em tarefas de diferentes domínios**. 2013. 52 f. Trabalho de Conclusão de Curso – Curso Superior de Ciências e Tecnologia – Universidade Federal Rural do Semi-Árido, Angicos, 2013.

SILVA, Pablo N. **Classificação Multirrótulo em Cadeia:** Novas Abordagens. 2015. 81 f. Dissertação (Mestrado em Computação) - Universidade Federal Fluminense, Niterói, 2014.

TSOUMAKAS, Grigorios; KATAKIS, Ioannis; VLAHAVAS, Ioannis. Random k-labelsets for multi-label classification. **IEEE Transactions on Knowledge and Data Engineering**, Los Alamitos, v. 99, 2010.

VALLIM, Rosane M. M. **Sistemas classificadores evolutivos para problemas multirrótulo**. 2009. 97 f. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Universidade de São Paulo, São Carlos, 2009.

VIEIRA, Sonia. **Introdução à Bioestatística**. 4. ed. Rio de Janeiro: Elsevier, 2011.

VILLANI, Leonardo. **Anotação automática de imagens médicas bidimensionais por meio de classificação multirrótulo**. 2013. 102 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal do ABC, Santo André, 2013.