Classificação do grau de acabamento de gordura da carcaça de bovinos de corte usando aprendizado de máquina

Higor Henrique Picoli Nucci

Classificação do grau de acabamento de gordura da carcaça de bovinos de corte usando aprendizado de máquina

Higor Henrique Picoli Nucci

Orientador: *Profº Drº Renato Porfirio Ishii*
Coorientador: *Drº Rodrigo da Costa Gomes*

Dissertação de mestrado entregue a Faculdade de Computação da Universidade Federal de Mato Grosso do Sul (FACOM-UFMS) como parte dos requisitos do Mestrado Profissional em Computação Aplicada.

UFMS - Campo Grande
maio/2019

Agradecimentos

Agradeço a minha amada esposa Ana Carolina Nucci por ter me apoiado a fazer esse mestrado e sempre me motivado a seguir em frente e perseguir meus sonhos.

Ao meu prezado orientador Renato Porfirio Ishii por ter me guiado ao longo dessa estrada e partilhar sua sabedoria comigo.

Ao meu coorientador Rodrigo da Costa Gomes pelo apoio e empenho na obtenção dos dados utilizados nesse trabalho e na interpretação dos mesmos.

Ao pesquisador da Embrapa Gelson Luís Dias Feijó pelo seu apoio na obtenção dos dados.

Ao professor Bruno Magalhães Nogueira pelas orientações com relação aos passos necessários para a criação de um modelo de Aprendizado de Máquina mais eficaz.

Por fim, agradeço a Superintendência de Gestão da Informação - SGI e a Secretaria de Estado de Meio Ambiente, Desenvolvimento Econômico, Produção e Agricultura Familiar - SEMAGRO pela liberação dos dados do programa Precoce MS para que este trabalho fosse possível.

Abstract

Nowadays, there is an increase in world demand for quality beef. In this way, programs to encourage livestock raising that produce a good carcass finish are becoming more frequent. The Government of the State of Mato Grosso do Sul has created an incentive program (Precoce MS) that stimulates producers to fit into production systems that lead to the slaughter of animals at young ages and superior carcass quality, towards a more sustainable production model. This work aims to build a classification model of carcass fatness degree using machine learning algorithms and to provide the cattle ranchers with indicators that help them to early finishing cattle with better carcass finishing. To achieve this goal, data from the Precoce MS program were used. The dataset contains twenty-nine different features with categorical data, discrete data and with 1.05 million cattle slaughter records. With the data in hand, the data mining process was initiated. In this process, the data were cleaned, transformed and reduced in order to extract patterns more efficiently. In the model selection step, the data was divided into five different data sets for performing cross-validation. The training set received 80% of the data and the test set received the other 20%, emphasizing that both had their data stratified respecting the percentage of each target class. The algorithms analyzed and tested in this work were Support Vector Machines, K-Nearest Neighbors, Naive Bayes and Random Forest Classifier. In order to obtain a better classification, the recursive feature elimination and grid search techniques were used in the models with the objective of selecting better characteristics and obtaining better hyperparameters, respectively. The precision, recall and f1 score metrics were applied in the test set to confirm the choice of the model, taking into account the confusion matrix. Finally, analysis of variance ANOVA indicated that there are significant differences between the models. The Tukey post-hoc test showed that the Random Forest Classifier and Support Vector Machines models are similar to a significance level of 5%. Therefore, these two classifiers can be used for the construction of a final model without prejudice

in the classification performance.

Keywords: Data mining, Early calf, Precoce MS, Precision Livestock

Resumo

Nos dias atuais, existe um aumento na demanda mundial por carne bovina de qualidade. Dessa forma, os programas de incentivo à criação de gado que produza bons acabamentos de carcaça vêm se tornando mais frequentes. O Governo do Estado de Mato Grosso do Sul criou um programa de incentivo (Precoce MS) que estimula produtores a se adequarem em sistemas de produção que acarretam no abate de animais em idade jovem e qualidade de carcaça superior, em direção a um modelo de produção mais sustentável. Sendo assim, surge a necessidade de propor formas de otimizar a engorda desses animais. Este trabalho tem por objetivo construir um modelo de classificação do grau de gordura da carcaça por meio de algoritmos de aprendizado de máquina e fornecer aos pecuaristas indicadores que os auxiliem a abater novilhos precoces com grau de gordura de carcaça cada vez melhor. Para alcançar esse objetivo, os dados do programa Precoce MS foram utilizados. O conjunto de dados contêm vinte e nove características diferentes com dados categóricos, discretos e com 1,05 milhão de registros de abates de bovinos. Com os dados em mãos, deu-se início ao processo de mineração de dados. Nesse processo, os dados foram limpos, transformados e reduzidos a fim de extrair padrões de forma mais eficiente. Na etapa de seleção do modelo, dividiu-se os dados em cinco conjuntos de dados diferentes para execução da validação cruzada. O conjunto de treinamento recebeu 80% dos dados e o conjunto de testes recebeu os outros 20%, ressaltando que ambos tiveram seus dados estratificados respeitando a porcentagem de cada classe alvo. Os algoritmos analisados e testados neste trabalho foram Support Vector Machines, K-Nearest Neighbors, Naive Bayes e Random Forest Classifier. A fim de obter uma melhor classificação, as técnicas *recursive feature elimination* e *grid search* foram utilizadas nos modelos com o objetivo de selecionar melhores características e obter melhores hiperparâmetros, respectivamente. As métricas *precision*, *recall* e *f1 score* foram aplicadas no conjunto de testes para confirmar a escolha do modelo, levando em consideração a matriz de confusão. Por fim, a análise de variân-

cia ANOVA indicou que existem diferenças significativas entre os modelos. O teste *post-hoc* de Tukey mostrou que os modelos Random Forest Classifier e Support Vector Machines são semelhantes a um nível de significância de 5%. Portanto, os dois classificadores podem ser usados para a construção de um modelo final sem prejuízo no desempenho da classificação.

Palavras-chave: Mineração de dados, Novilho precoce, Precoce MS, Pecuária de precisão

Lista de Figuras

1.1 Maturidade dentária de novilhos precoces. 3
1.2 Graus de gordura do acabamento de carcaças. 4

2.1 Formas de pré-processamento de dados. 11
2.2 Representação visual de um processo de aprendizado de máquina supervisionado. 15
2.3 Conjunto de dados de treinamento usando validação cruzada *k-fold*. 18
2.4 Modelo básico de uma árvore de decisão aplicada no conjunto de dados iris. 20
2.5 Mapeamento de um conjunto de dados usando K-Nearest Neighbors. 21
2.6 Exemplo do uso do algoritmo SVM linear. 23
2.7 Mapeamento de um conjunto de dados usando Support Vector Machines usando *kernel*. 23

3.1 Fluxograma completo da metodologia utilizada. 27
3.2 Distribuição do grau de gordura da carcaça. 33
3.3 Comparação da distribuição das classes com balanceamento do conjunto de dados usando técnicas *under-sampling*, *over-sampling* e híbridas. 34
3.4 Visualização do processo de validação cruzada com o conjunto de treinamento balanceado. 38

4.1 Resultados normalizados da predição dos algoritmos no conjunto de dados de testes, com a representação em escala de 0,00 a 1,00, para análise de erro. 40
4.2 Características mais relevantes de acordo com o algoritmo RFC e a técnica de seleção de características RFE. 42

4.3 Resultados do algoritmo Random Forst Classifier quando executado sobre o conjunto de testes. 43
4.4 Comparação das médias de desempenho da métrica *f1 score* de cada modelo. 44
4.5 Esquema proposto da aplicação do modelo baseado em serviços Web. 45

Lista de Tabelas

1.1 Esquema simplificado da classificação de carcaças bonificadas pelo programa Precoce MS. 4

2.1 Exemplo modificado de um conjunto de dados com dados ausentes. 12
2.2 Exemplo modificado de um conjunto de dados com dados categóricos. 17
2.3 Conjunto de dados categóricos após a conversão para dados numéricos. 17

3.1 Cinco amostras aleatórias dos dados cadastrais de um estabelecimento rural. 28
3.2 Perguntas que não classificam o processo produtivo de um estabelecimento rural, seus respectivos rótulos no conjunto de dados e seus possíveis valores. 29
3.3 Perguntas que classificam o processo produtivo de um estabelecimento rural, seus respectivos rótulos no conjunto de dados e seus possíveis valores. 29
3.4 Cinco amostras aleatórias do conjunto de dados dos abates de bovinos. 30
3.5 Classes para a classificação do grau de gordura da carcaça. . . . 31
3.6 Descrição do conjunto de dados completo após ser pré-processado. 32
3.7 Dois exemplos de matrizes de confusão. 35
3.8 Exemplos de verdadeiros positivos, falsos negativos, falsos positivos e verdadeiros negativos. 35

4.1 Comparação da média da *accuracy* entre os quatro modelos escolhidos. 39
4.2 Comparação das métricas *Precision*, *Recall* e *F1 score* para cada uma das 5 classes, com suas respectivas médias, entre os modelos. 41

4.3 Comparação da média da *acurácia* e *f1 score* entre os quatro modelos escolhidos após a otimização com *grid search*. 42

Conteúdo

Lista de Figuras x
Lista de Tabelas xii
Sumário xiv

1 Introdução **1**
1.1 Contextualização 1
1.2 O problema 5
1.3 Objetivos 6
1.4 Solução 6
1.5 Estrutura do documento 7

2 Referencial Teórico **9**
2.1 Mineração de dados 9
2.1.1 Pré-processamento 10
2.2 Aprendizado de Máquina 14
2.2.1 Aprendizado supervisionado 15
2.2.2 Classificação 16
2.2.3 Dados categóricos 16
2.2.4 Validação cruzada 18
2.3 Algoritmos de aprendizado de máquina supervisionado 19
2.3.1 Random Forest Classifier 19
2.3.2 Naive Bayes 20
2.3.3 K-Nearest Neighbors 21
2.3.4 Support Vector Machines 22
2.4 Ferramentas utilizadas 24
2.4.1 Scikit-Learn 24
2.5 Trabalhos relacionados 24

3 Materiais e métodos **27**
3.1 Abordagem 27

3.2 Aquisição dos dados 28
3.3 Pré-Processamento 30
3.4 Pré-Análise 32
3.5 Métricas de desempenho 34
3.6 Seleção do modelo 37

4 Resultados e discussões **39**
4.1 Aplicação do modelo 45

5 Conclusões **47**
5.1 Resumo dos Objetivos e Principais Resultados 47
5.2 Dificuldades encontradas 48
5.3 Trabalhos Futuros 48

Referências **59**

CAPÍTULO

1

Introdução

1.1 Contextualização

O Brasil ocupa uma posição de destaque no cenário mundial de produção de carne bovina. Apenas no terceiro trimestre de 2018, foram abatidas 8,28 milhões de cabeças de bovinos, que estavam sendo supervisionados por algum serviço de inspeção sanitária, como mostram os dados do relatório de Estatística da Produção Pecuária, fornecido pelo Instituto Brasileiro de Geografia e Estatística [1]. Esses dados são 3,7% maiores que o mesmo período do ano imediatamente anterior.

O aumento de produção no setor agropecuário brasileiro, como mostrado por meio de dados da Secretaria de Comércio Exterior (Secex), no terceiro trimestre de 2018, é tanto no faturamento ($20,80\%$) como em volume ($25,40\%$) das exportações de carne bovina *in natura*, comparado ao mesmo período no ano anterior. Sendo assim, o Ministério da Agricultura, Pecuária e Abastecimento estima que, até 2020, a produção nacional de carnes poderá suprir 44,5% do mercado mundial, mantendo o Brasil em primeiro lugar no ranking de exportações de carnes [2].

Portanto, para se manter no topo do ranking mundial de exportações, o governo brasileiro, associações de produtores e de raças e frigoríficos têm se empenhado em criar programas que incentivem a produção de carnes com qualidade superior à média brasileira. Um programa criado para este fim e que merece destaque é o de bonificação de carcaças bovinas[1], instituído pela

[1]Entende-se por carcaça o bovino abatido, sangrado, esfolado, eviscerado, desprovido de cabeça, patas, rabada, glândula mamária (na fêmea), verga, exceto suas raízes, e testículos (no macho). Após sua divisão em meias carcaças retiram-se ainda os rins, gorduras perirrenal

normativa 9 de 04 de maio de 2004 do Ministério da Agricultura, Pecuária e Abastecimento [4] que instaura em todo o território nacional, o Sistema Brasileiro de Classificação de Carcaças de Bovinos (SBCCB), a ser implantado nos estabelecimentos de abate sob Serviço de Inspeção Federal (SIF), tendo como base as características indicativas de qualidade: sexo e maturidade do animal, peso e acabamento da carcaça.

Logo, para gerar uma carcaça bovina que contribua para uma carne de maior qualidade, ou seja, que possa ser comercializada com um valor superior, algumas características específicas quanto às tipificações devem ser observadas. Segundo Felício [5], a carcaça bovina pode ser tipificada pelas seguintes características do animal e de sua carcaça:

- Classificação por gênero;
- Idade aproximada (maturidade óssea ou dentária);
- Faixa de peso;
- Conformação e acabamento, ambos avaliados na carcaça quente, ainda na sala de matança; e
- Outros como cor da carne, *marbling* (mármore) e área do olho de lombo, por exemplo, que só são aferíveis depois do resfriamento das carcaças, quando terá ocorrido o *rigor mortis*, ou seja, os músculos transformam-se em carne por meio de diversas modificações bioquímicas e físicas.

No estado de Mato Grosso do Sul, foram lançados o decreto Nº 14.526, de 28 de julho de 2016 e a resolução conjunta SEFAZ/SEPAF Nº 69 de 30 de agosto de 2016 que fizeram alterações ao programa de bonificação para novilhos precoces[2] já existente e instituiu o programa Precoce MS, sendo esse gerido pelo Programa de Avanços na Pecuária de Mato Grosso do Sul (Proape). Dentre as alterações aplicadas, pode-se citar como mais impactante a inclusão do processo produtivo da fazenda no cálculo para o incentivo fiscal.

O programa Precoce MS bonifica o produtor com retorno financeiro de até 67% do ICMS recolhido por carcaça de um animal abatido e classificado como novilho precoce. O cálculo da porcentagem do valor do ICMS a ser devolvido ao produtor leva em consideração o acabamento da carcaça e as características do processo produtivo da fazenda, considerando-se uma proporcionalidade de 70% e 30%, respectivamente.

---

e inguinal, “ferida de sangria”, medula espinhal, diafragma e seus pilares [3].

[2] O termo novilho precoce foi definido pela Portaria Nº 268, de 4 de maio de 1995 [6], como bovino jovem que gera um tipo de carcaça de boa qualidade e que atenda as especificações da Portaria nº 612, de 5 de outubro de 1989 [7].

A classificação de um animal como novilho precoce é realizada por meio dos seguintes parâmetros: gênero (F = fêmea; C = macho castrado; e M = macho inteiro) e a maturidade dentária (Figura 1.1). A fim de determinar a tipificação da carcaça, quanto ao seu acabamento, é adotado como parâmetro a medida da gordura subcutânea, como mostrado na Fig. 1.2. Por fim, o peso do animal antes do abate deve ser no mínimo 180 quilogramas para fêmeas e 225 quilogramas para machos (inteiros ou castrados) e pelo menos 60% do lote a ser abatido deve ser composto por novilhos precoces [5].

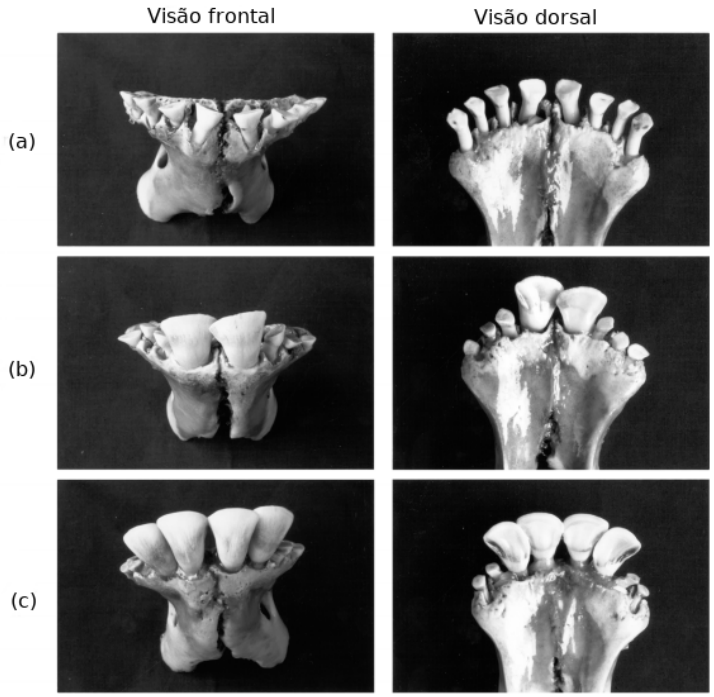

Figura 1.1: Maturidade dentária de novilhos precoces: (a) J0 = apenas dentes de leite; (b) J2 = dois dentes incisivos permanentes; e (c) J4 = quatro dentes incisivos permanentes. Fonte: [8].

Os processos produtivos de uma propriedade, segundo a resolução Conjunta SEFAZ/SEPAF Nº 69 DE 30 de agosto de 2016 [10], são avaliados valorizando as propriedades que atendam aos seguintes critérios:

1. utilizem ferramentas que permitam a gestão sanitária individual de bovinos;

2. apliquem regras e conceitos de boas práticas agropecuárias;

3. apliquem tecnologias que promovam a sustentabilidade do sistema produtivo, em particular aquelas que visem à mitigação da emissão de carbono por meio de práticas de agropecuária de baixo carbono;

4. participem de associações de produtores visando à produção comercial sistematizada e organizada conforme padrões pré-estabelecidos para atendimento de acordos comerciais.

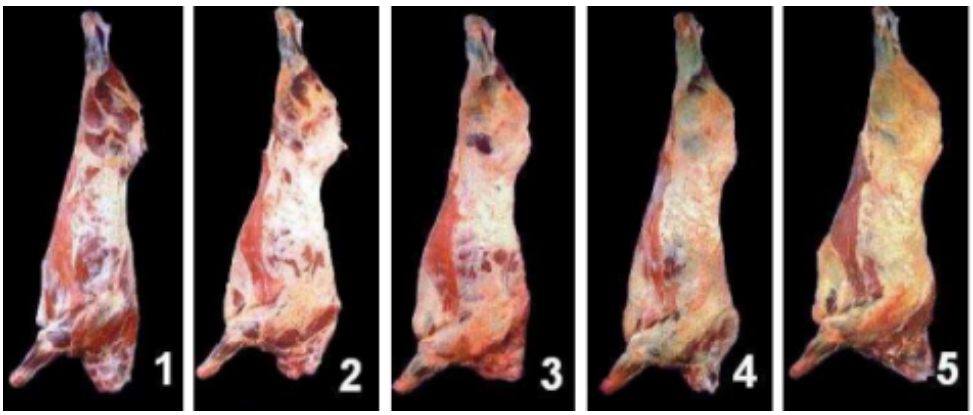

Figura 1.2: Graus de gordura do acabamento de carcaças: 1 = Magra - Gordura ausente; 2 = Gordura Escassa - 1 a 3 mm de espessura; 3 = Gordura Mediana - acima de 3 e até 6mm de espessura; 4 = Gordura Uniforme - acima de 6 e até 10 mm de espessura; e 5 = Gordura Excessiva - acima de 10 mm de espessura. Fonte: [9].

Esses critérios avaliam o processo produtivo da propriedade. Sendo assim, um estabelecimento pode ser classificado como **Simples**, **Intermediário** e **Avançado**. Estabelecimentos rurais são classificadas como **Simples** se atendem a nenhum ou ao menos um dos critérios. Os que são classificados como **Intermediário** devem atender pelo menos dois critérios. Por fim, um estabelecimento classificado como **Avançado** deve atender no mínimo três critérios.

Dessa forma, a Tabela 1.1 simplificada pode ser usada para exemplificar as regras do programa. As três primeiras colunas mostram os dados do animal abatido e representam até 70% do valor do incentivo. As colunas restantes referem-se aos dados da propriedade avaliada e representam até 30% do valor do incentivo. Sendo assim, o valor do incentivo a ser retornado pelo frigorífico ao produtor é a porcentagem que ele atingir na tabela (até o máximo de 67%) sobre o valor do ICMS. O restante do valor do ICMS é pago ao Estado pelo frigorífico.

70% Produto			30% Processo produtivo		
Tipificação	Maturidade	Acabamento	Avançado 30%	Intermediário 26%	Simples 21%
M, C, F	J0	3, 4	67	64	61
M, C, F	J2	3, 4	62	59	56
C, F	J4	3, 4	48	45	42
M, C, F	J0	2	62	59	56
M, C, F	J2	2	39	36	33
C, F	J4	2	22	19	16

Tabela 1.1: Esquema simplificado da classificação de carcaças bonificadas pelo programa Precoce MS [11] [10]. Tipificação: F = fêmea, C = macho castrado e M = macho inteiro; Maturidade: J0 = apenas dentes de leite, J2 = dois dentes incisivos permanentes e J4 = quatro dentes incisivos permanentes.

1.2 O problema

A indústria de pecuária de corte brasileira, principalmente os produtores, vivem um momento de muita pressão, devido a um mercado cada vez mais exigente com relação a qualidade da carne e remunerando melhor por animais cada vez mais novos (novilhos precoces). De todos os abates no terceiro trimestre de 2018, foram gerados um total de 2.106.195 toneladas de peso de carcaças, totalizando 426.089 toneladas de carne *in natura* exportadas [1].

Independentemente do Brasil ser o líder mundial em quantidade de carne bovina *in natura* exportada, o rendimento financeiro é relativamente baixo. Visto que não exporta para os mercados que mais pagam, pois a carne nacional não atende critérios para alguns mercados que pagam mais [12].

Nesse contexto, existem fatores na bovinocultura que podem influenciar na qualidade geral da carne produzida. Um fator fundamental é uma alimentação de alta qualidade e contínua para o rebanho [13]. Entretanto, um dos problemas na produção de novilhos precoces pode ser o custo com essa alimentação de excelência [14].

Uma menor idade ao abate tem sido relacionada com maior lucratividade das propriedades rurais [15]. Levando em consideração a sustentabilidade econômica da atividade, o estímulo à adoção do modelo precoce de produção pode contribuir para a sustentação da atividade. Existem outros fatores que também são considerados importantes, tais como, a capacidade genética e a maturidade, visto que animais jovens e de boa genética possuem uma eficiência maior em converter comida consumida em ganho de peso e gordura, o que refletirá diretamente no rendimento geral da carcaça [16] [17] [18].

Do ponto de vista econômico, alcançar os fatores que influenciam na qualidade final da carne eleva os custos operacionais do produtor. Dentre esses fatores destacam-se: obter um gado com boa genética, proporcionar uma alimentação eficaz e garantir um tempo adequado para engorda [16]. Do ponto de vista tecnológico, existe um amplo espaço para a melhoria na produção de gado de corte por meio da bovinocultura de precisão. Sendo assim, tecnologias que resultem no aumento da probabilidade de sucesso econômico são cruciais. Posto que o produtor terá acesso a dados que oferecerão suporte à tomada de decisão mais eficaz, que otimize sua produção e seu equilíbrio econômico [19] e o conduza a produzir animais com melhor adequação aos critérios de carcaça, com menor custo de produção e maior obtenção de bonificações.

1.3 Objetivos

O objetivo principal deste trabalho é construir um modelo de classificação do acabamento da carcaça por meio de algoritmos de aprendizado de máquina. O conjunto de dados utilizado contêm os dados de abate de bovinos cadastrados no programa estadual Precoce MS e do processo produtivo de cada estabelecimento rural correspondente. Esse modelo de classificação apoiará os produtores da bovinocultura de corte na tomada de decisão, visando aumentar a qualidade da carne produzida.

Os objetivos específicos são os que seguem:

1. Testar algoritmos de aprendizado de máquina para descobrir qual deles pode gerar uma classificação com maior acurácia;
2. Descobrir quais as características do conjunto de dados possuem maior relevância em relação ao grau acabamento de gordura da carcaça;

1.4 Solução

Neste trabalho é proposta a análise dos dados do programa Precoce MS por meio de algoritmos de Aprendizado de Máquina. Esta abordagem tem por objetivos auxiliar os produtores de bovinos como uma forma de apoio à tomada de decisão para elevar a qualidade da carne e, por conseguinte, aumentar suas receitas com maiores bonificações.

A abordagem é dividida em quatro etapas, sendo elas:

1. Aquisição dos dados do programa Precoce MS;
2. Pré-processamento dos dados para transformá-los em dados numéricos;
3. Aplicação dos algoritmos de Aprendizado de Máquina para a seleção do modelo que apresentar melhor acurácia;
4. Análise do classificador a fim de melhorar seu desempenho tanto na acurácia quanto em métricas como *precision*, *recall* e *f1 score*.

Na etapa de aquisição de dados foi feita a negociação com os órgãos governamentais responsáveis pelos dados. Eles disponibilizaram um conjunto de dados com os dados do processo produtivo e outro com os abates de bovinos de todos os estabelecimentos rurais cadastrados no programa Precoce MS. O conjunto de dados de abates de bovinos contêm um pouco mais de 1,1 milhão de registros, no período de 09/02/2017 até 23/01/2019, e o outro contêm 1.595 estabelecimentos rurais e seus respectivos processos produtivos.

A segunda etapa faz o pré-processamento dos dados com o objetivo de gerar um conjunto de dados unificado somente com dados numéricos. Sendo assim, essa etapa inicia com a junção dos dois conjuntos por meio do identificador do estabelecimento rural. O primeiro passo do pré-processamento foi a transformação das características categóricas em valores numéricos por meio de técnicas de transformação de dados. O segundo passo do pré-processamento foi a normalização dos valores numéricos a fim de expressar os valores em uma distribuição mais regular, tal como $[0,0;1,0]$. O conjunto de dados resultante conta com vinte e nove características diferentes e 1,05 milhão de registros.

Na etapa de seleção do modelo, os algoritmos analisados e testados neste trabalho, usando validação cruzada com cinco dobras, foram Support Vector Machines, K-Nearest Neighbors, Naive Bayes e Random Forest Classifier. Os modelos de classificação que apresentaram o melhor acurácia foram: Random Forest Classifier (70,45%$(+/-)0,139$) e Support Vector Machines (70,11%$(+/-)0,003$).

Portanto, este trabalho contribui a criação de um processo de Mineração de Dados e construção de um modelo de Aprendizado de Máquina capaz de auxiliar os produtores rurais na diminuição das incertezas relacionadas à adequação do processo produtivo e, consequentemente, auxiliá-los nas tomadas de decisões.

1.5 Estrutura do documento

O presente trabalho está organizado em cinco capítulos. Neste capítulo foram definidos o contexto, o problema, a solução abordada e os objetivos gerais e específicos. O Capítulo 2 apresenta o Referencial Teórico, onde trata os conceitos base sobre Mineração de Dados, Aprendizado de Máquina, seus algoritmos, algumas ferramentas utilizadas e os trabalhos relacionados a essa dissertação. No Capítulo 3, são retratados os materiais e metodologia utilizados para fazer a classificação do grau de gordura da carcaça dos bovinos abatidos pelo programa Precoce MS. O Capítulo 4 mostra os resultados dos testes com os algoritmos de Aprendizado de Máquina. Por fim, no Capítulo 5 são apresentadas as conclusões, dificuldades encontradas e trabalhos futuros propostos.

CAPÍTULO

2

Referencial Teórico

2.1 Mineração de dados

O termo Mineração de Dados, por ser genérico e representar um assunto multidisciplinar, é encontrado em muitas literaturas como Extração de Conhecimento de Base de Dados ou Knowledge Discovery from Data (KDD), em inglês. Sendo assim, existe uma definição mais realista e atual:

> "Extração de Conhecimento de Base de Dados é o processo de identificação de padrões válidos, novos e potencialmente úteis e compreensíveis embutidos nos dados [20]."

O processo de Mineração de Dados pode ser descrito como um conjunto de atividades a serem realizadas sequencialmente. São elas:

1. **Limpeza dos dados:** etapa na qual removem-se a sujeira e os dados inconsistentes;

2. **Integração dos dados:** etapa em que conjuntos de dados podem ser combinados;

3. **Redução de dados:** momento no qual dados redundantes e sem valor analítico são eliminados;

4. **Transformação dos dados:** fase em que os dados são transformados e consolidados em formatos apropriados para mineração de dados, por meio da realização de operações de condensação ou agregação;

5. **Mineração de dados:** processo essencial em que algoritmos inteligentes são aplicados para extrair padrões;

6. **Avaliação de padrões:** etapa que visa identificar padrões realmente valiosos para o contexto;

7. **Representação do conhecimento:** etapa final na qual técnicas de visualização e de representação do conhecimento são usadas para apresentar os dados aos usuários.

Todos esses processos podem ser divididos em três grandes passos: **Pré-processamento**, etapas de 1 a 4, **Extração de Padrões**, etapas 5 e 6, **Pós-processamento**, etapa 7.

Dessa forma, pode-se classificar a Mineração de Dados como sendo o processo de descoberta do conhecimento e de padrões interessantes para um determinado contexto [21].

2.1.1 Pré-processamento

A etapa de Pré-Processamento de Dados é uma série de atividades (Figura 2.1) executadas ordenadamente, que resultará em uma espécie de arranjo de informações. Inicialmente, são coletadas informações, ou dados, que passam por uma organização e que no final serão úteis para o objetivo do usuário ou do sistema que pretende utilizá-las.

A qualidade dos dados e a quantidade de informações úteis que o mesmo contém são fundamentais para determinar o quão bem um algoritmo de aprendizagem de máquina pode aprender. Portanto, é absolutamente necessário que os dados sejam examinados e pré-processados para um conjunto de dados novo antes de usá-lo em um algoritmo de aprendizagem.

Limpeza dos dados

Em alguns casos, os dados são recebidos em formatos de linhas e colunas, como mostrado na Tabela 2.1 e é muito comum que existam um ou mais dados ausentes na amostragem, como pode ser observado na linha com *index* = 5 e coluna *sepal_width*. Esses dados podem ter sido perdidos na fase de coleta ou devido a algum fator inesperado. Trabalhar com um conjunto de dados dessa forma é desafiador, pois os sistemas computacionais, em geral, não têm habilidade para manipular esse tipo de informação de forma eficaz. Então, é imprescindível que os dados ausentes em uma amostra sejam tratados antes de utilizá-los em algum modelo de classificação [22].

Excluir a coluna (característica) ou a amostra (linha) com os dados ausentes do conjunto de dados é uma das maneiras mais fáceis de lidar com esse

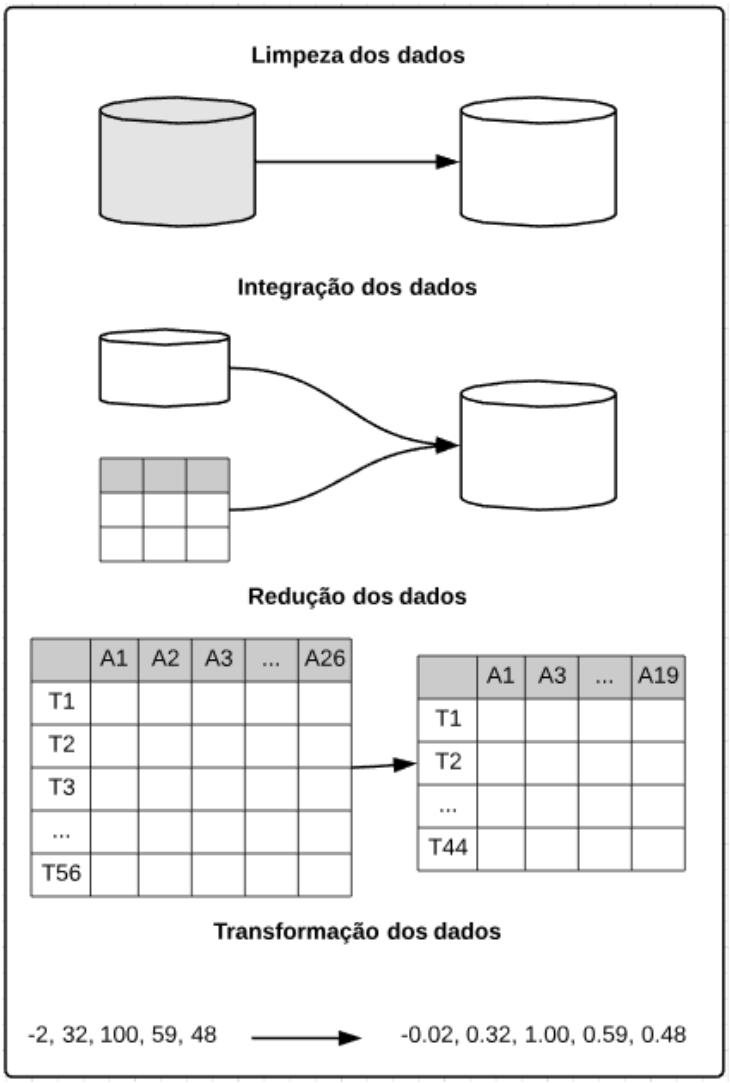

Figura 2.1: Formas de pré-processamento de dados.

tipo de problema. Entretanto, remover dados parece ser uma abordagem conveniente, mas apresenta desvantagens. Uma delas é que o conjunto de dados pode ser tornar tão pequeno, ou sem informações relevantes, que a aplicação de algum algoritmo de aprendizado de máquina para obter uma predição valiosa seja impossível.

Geralmente, o uso de técnicas de interpolação para lidar com dados ausentes sem precisar removê-los é indicado. Uma das técnicas mais comuns é o uso de *imputação média*, que consiste em simplesmente trocar os valores ausentes pela média dos valores da coluna inteira. Outra forma é trocar os valores pela mediana ou pelo mais frequente da coluna. Este último é mais usado no caso dos dados serem categóricos, ou seja, não numéricos [24].

Por fim, um problema comum em conjuntos de dados é a sujeira ou, em inglês, como é mais encontrado, *outlier*. A sujeira é classificada como sendo um erro aleatório ou a variação muito distante de uma medida com relação as outras. Nesse caso, deve-se "suavizar os dados" a fim de encontrar uma

index	sepal_length	sepal_width	petal_length	petal_width	class
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	7.0	3.2	4.7	1.4	Iris-versicolor
5	6.3		6.0	2.5	Iris-virginica
6	5.8	2.7	5.1	1.9	Iris-virginica

Tabela 2.1: Exemplo modificado do conjunto de dados Iris [23] com dados ausentes.

medida que mantenha o valor daquela amostra mais próxima das outras.

Em mineração de dados, existem três técnicas para a remoção de *outliers* [21]. A primeira delas é trocar a sujeira pela média dos valores. A segunda é trocar a sujeira pela mediana dos valores. A terceira é trocar os valores pela fronteira, ou seja, deve-se encontrar o valor mínimo e máximo para determinada característica e substituir a sujeira pelo valor mais próximo à fronteira. No geral, quanto maior for a distância da sujeira com relação aos outros valores, maior será a "suavização dos dados".

Integração dos dados

Constantemente, os dados necessários para análise são encontrados em fontes de dados diferentes. Nesse caso, a integração dos dados em uma única fonte de dados pode ajudar a evitar redundância, melhorar a acurácia e aumentar a velocidade com que os dados são processados.

Alguns tipos de redundâncias podem ser detectados por meio de uma análise de correlação. Dadas duas características, essa análise pode medir o quão forte uma característica implica sobre a outra. No caso de dados numéricos, pode-se usar o coeficiente de correlação (também conhecido como **coeficiente de Pearson**), que mostra o quão um valor A pode variar com relação a outro valor B [21].

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B} \tag{2.1}$$

A Equação 2.1 mostra como o coeficiente de Pearson é calculado, em que n é a quantidade de amostras, a_i e b_i são, respectivamente, os valores de A e B na amostra i, $\bar{A}$ e $\bar{B}$ são as médias dos valores de A e B, σ_A e σ_B são os desvios padrões de A e B e $\sum_{i=1}^{n}(a_i b_i)$ é o somatório do produto cruzado de AB.

O resultado da Equação 2.1 é um valor entre 1 e -1. Se o resultado é igual a 0, então A e B são independentes e não existe correlação entre eles. Se o resultado é menor que 0, então A e B são negativamente correlacionados, ou seja, quando um valor aumenta o outro diminui. Por fim, se o resultado é

maior que 0, então A e B são positivamente correlacionados, ou seja, o valor de A aumenta assim como o valor de B também aumenta.

Outra preocupação no momento da integração de dados é a duplicação de características, que deve ser evitada para não aumentar a redundância dos dados. O uso de fontes de dados não normalizadas pode implicar em características com mesmo nome e valores. Dessa forma, deve-se remover as colunas duplicadas.

Redução dos dados

A técnica de redução dos dados é utilizada para obter uma representação do conjunto de dados com um volume menor e, ainda assim, manter a integridade do mesmo. Consequentemente, o processo de mineração de dados em um conjunto de dados reduzido deve ser mais eficiente e produzir o mesmo resultado analítico que o não reduzido. Existem três estratégias principais para lidar com a redução de dados: redução da dimensionalidade, redução da numerosidade e compressão dos dados.

Redução da dimensionalidade é o processo de diminuir o número de variáveis aleatórias ou atributos em consideração. **Redução da numerosidade** consiste em substituir o volume de dados original por formas alternativas e menores da representação dos dados podendo ser paramétricas ou não-paramétricas. Em **compressão de dados** transformações são aplicadas afim de obter uma representação reduzida ou comprimida dos dados originais. Nessa última, a técnica é considerada sem perda (*lossless*) caso os dados possam ser comprimidos sem perder qualquer informação. No contrário, a técnica é considerada com perdas (*lossy*) no caso de ser possível apenas reconstruir um conjunto de dados aproximado do original [21].

Transformação dos dados

Os dados são transformados ou consolidados para que o processo de mineração de dados possa ser mais eficiente e os padrões encontrados sejam mais fáceis de entender [21]. Uma estratégia para a transformação de dados utilizada no contexto desta dissertação é a **normalização**.

A unidade de medida usada para determinada característica pode afetar negativamente a análise dos dados. A técnica de normalização consiste em expressar os valores de uma característica em unidades de medida menores. Dessa forma, os dados são transformados em uma distribuição mais regular, tais como $[-1, 1]$ ou $[0.0, 1.0]$. Para normalizar os dados, pode-se usar o método *min-max*. Esse método realiza a transformação linear dos dados originais (Equação 2.2) em que min_A e max_A são os valores mínimo e máximo de uma característica A e mapeia um valor v_i', dessa mesma característica, na

distribuição $[novo\_max_A, novo\_min_A]$.

$$v_i^{'} = \frac{v_i - min_A}{max_A - min_A}(novo\_max_A, novo\_min_A) + novo\_min_A \tag{2.2}$$

2.2 Aprendizado de Máquina

A área de Aprendizado de Máquina é uma frente de estudo da Inteligência Artificial que permite a um sistema computacional tornar-se mais assertivo em prever eventos sem que seja explicitamente programado [25]. A premissa básica da aprendizagem em máquina é construir algoritmos específicos que recebam dados de entrada e os use por meio de análises estatísticas e matemáticas para tentar prever um valor de saída dentro de um intervalo considerado aceitável.

Um agente é dito estar aprendendo quando melhora o seu desempenho em suas tarefas após realizar as suas observações sobre o ambiente externo. Esse agente pode reaprender se houver uma realimentação adequada. Logo, o autor Tom M. Mitchel [26] oferece uma das definições de aprendizagem de máquina mais assertiva e atual:

> "Um programa de computador é dito aprender com a experiência *E* em relação a alguma tarefa *T* e alguma medida de desempenho *D*, se seu desempenho em *T*, conforme medido por *D*, melhora com a experiência *E*."

Ao contrário de requisitar pessoas para obter regras e construir modelos manualmente usando grandes quantidades de dados, o aprendizado de máquina oferece uma alternativa mais eficiente para capturar o conhecimento já adquirido para melhorar gradualmente o desempenho de modelos preditivos e tomar decisões orientadas por dados.

Não só o aprendizado de máquina é cada vez mais importante na pesquisa em computação, mas também desempenha um papel cada vez maior em nossa vida cotidiana. Nos tempos atuais, vivemos em uma era de abundância de informação e utilizar algoritmos de autoaprendizado para transformar dados em conhecimento, a fim de criar predições, é um diferencial muito importante para qualquer setor da indústria.

Existem quatro tipos diferentes de modelos de aprendizado de máquina: aprendizado supervisionado, aprendizado semi-supervisionado, aprendizado não supervisionado e aprendizado por reforço [27] [28]. O contexto deste trabalho tratará apenas do modelo de aprendizado de máquina supervisionado.

2.2.1 Aprendizado supervisionado

Os modelos mais comuns de algoritmos de aprendizado de máquina são os supervisionados. Em um modelo supervisionado, todos os dados do conjunto de dados já estão mapeados e classificados. Os algoritmos supervisionados aprendem um modelo a partir de dados de treinamento rotulados que nos permitam prever dados não vistos ou futuros.

Um conjunto de entrada que deve ser processado é uma coleção de características que já foram quantificadas e medidas por algum mecanismo externo. Assim como apresentado na Figura 2.2, o termo supervisionado refere-se a um conjunto de amostras nas quais as características de entrada X e os alvos de saída Y desejados já são conhecidos e contêm os respectivos rótulos [22].

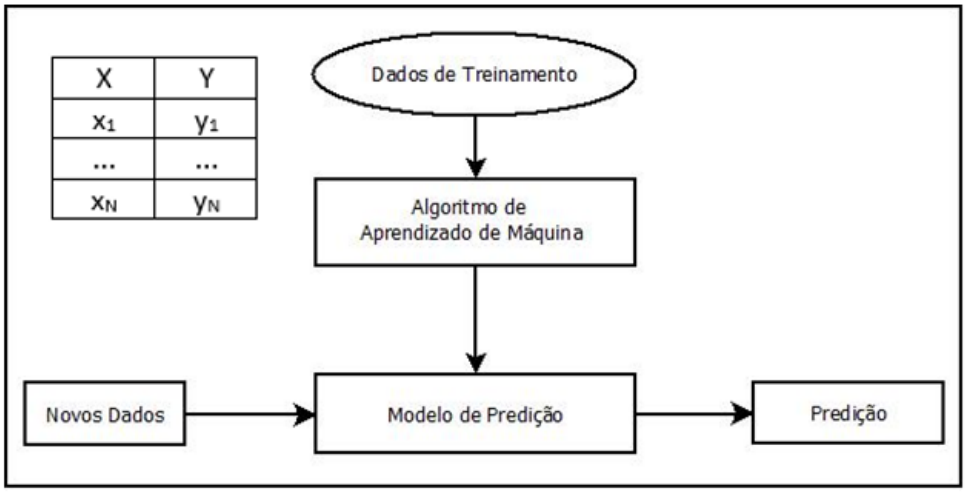

Figura 2.2: Representação visual de um processo de aprendizado de máquina supervisionado.

Desta maneira, um algoritmo que usa aprendizagem supervisionada aprende por meio da observação de exemplos de conjunto de pares de entrada e saída, como demostrado pela Equação 2.3.

$$(x_1, y_1), (x_2, y_2), ...(x_N, y_N) \tag{2.3}$$

Logo, o resultado é uma hipótese h que mapeia uma entrada desconhecida para uma saída que esteja mais próxima de uma função real f [27].

$$y = f(x) \tag{2.4}$$

Portanto, observando um conjunto de dados de amostra X, pode-se encontrar uma hipótese h que possa classificar uma saída y baseado em novos dados de entrada, como apresentado na Equação 2.4.

2.2.2 Classificação

O objetivo na classificação é ler dados de entrada x e atribuí-lo a uma das classes de saída y baseado em observações passadas. No cenário mais comum, as classes são consideradas disjuntas, de modo que cada entrada é atribuída a uma única classe. Essas classes devem ser discretas, finitas e desordenadas e podem ser consideradas como membros de um mesmo grupo. O espaço entre a regiões do grupo são denominados de limites de decisão.

A classificação que, para um conjunto de entrada $X = \{x_1, x_2, ..., x_n\}$, pode ter apenas dois tipos de saída, é denominada de classificação binária. Neste modelo de classificação, o resultado y esperado para as C categorias de uma predição é uma representação binária $y \in \{0, 1\}$. Sendo que $y = 0$ representa a classe C_1 e $y = 1$ a classe C_2. Pode-se interpretar o valor de y como a probabilidade de que a classe seja C_1, com os valores de probabilidade tomando apenas os valores extremos de 0 e 1.

No entanto, existem entradas que podem ser classificadas em mais de dois tipos e estas são denominadas de classificação multiclasse. Dessa forma, é conveniente usar um esquema de codificação em que y é um vetor de comprimento C, de modo que, se a classe for C_j, todos os elementos y_k de y são zero, exceto o elemento y_j, o que leva o valor 1. Por exemplo, se tivermos $C = 5$ classes, então um padrão da classe 2 receberia o vetor alvo da Equação 2.5.

$$y = (0, 1, 0, 0, 0) \tag{2.5}$$

Sendo assim, podemos interpretar o valor de y_k como sendo a porcentagem de a classe ser C_k [29].

2.2.3 Dados categóricos

Os dados categóricos são conhecidos por ocultar e mascarar muitas informações interessantes em um conjunto de dados. Em exemplos do mundo real, é muito comum encontrar conjuntos de dados que contêm não apenas valores numéricos, mas também, valores categóricos, ou seja, que estão em formato de texto, como apresentado na coluna *buying* da Tabela 2.2.

Muitos algoritmos e ferramentas de aprendizado de máquina trabalham apenas com características numéricas. Sendo assim, é crucial usar técnicas para lidar com esses valores categóricos, ao invés de remover uma característica ou uma amostra do conjunto de dados que será usado para a entrada do algoritmo de predição.

O primeiro passo para iniciar a conversão dos dados, é saber se eles podem ser classificados como *ordinal* ou *nominal*. Características ordinais podem ser entendidas como as que são de ordem de grandeza. Por exemplo, a caracterís-

index	buying	maint	doors	persons	lug_boot	safety	class
1	vhigh	vhigh	2	2	small	low	unacc
2	vhigh	vhigh	2	2	small	med	unacc
3	vhigh	vhigh	2	2	small	high	unacc
4	med	high	2	4	med	high	acc
5	med	high	2	4	big	med	acc
6	med	med	4	4	med	high	vgood

Tabela 2.2: Exemplo modificado de um conjunto de dados com dados categóricos [30].

tica *safety* da Tabela 2.2 pode ter seus possíveis valores representados como $low < med < high$. Em contraste, as características nominais são apenas texto que não tem relação direta entre seus valores. Por exemplo, a característica *persons* da Tabela 2.2 que pode receber os valores $\{2, 4, more\}$.

Na conversão de características ordinais pode-se usar uma técnica chamada *label encoder*. Trata-se de atribuir valores numéricos para os rótulos de acordo com sua ordem de grandeza, ou seja, estão sempre entre 0 e $C - 1$. No exemplo citado, o resultado para a característica *safety* seria $high = med + 1 = low + 2$.

Por outro lado, para converter características nominais pode-se usar a técnica *dummy coding* e convertê-los em novas características duplicadas que representam cada uma das características atuais. Nessa técnica, a presença de um nível é representada por 1 e a sua ausência por 0. Usando o exemplo da característica *persons*, o resultado seria a remoção dessa coluna e a criação de três novas características falsas (*dummy*) $\{$*persons_2*, *persons_4*, *persons_more*$\}$ [24].

Aplicando essas técnicas no nosso exemplo, teremos como resultado a Tabela 2.3 normalizada.

buying	maint	doors	persons_2	persons_4	persons_more	lug_boot	safety	class
3	3	2	1	0	0	0	0	unacc
3	3	2	1	0	0	0	1	unacc
3	3	2	1	0	0	0	2	unacc
1	2	2	0	1	0	1	2	acc
1	2	2	0	1	0	2	1	acc
1	1	4	0	1	0	1	2	vgood

Tabela 2.3: Conjunto de dados categóricos após a conversão para dados numéricos.

Um ponto negativo da conversão dos dados categóricos é que, se tiverem muitos níveis, pode-se prejudicar o desempenho geral do algoritmo de predição.

2.2.4 Validação cruzada

Para medir a precisão e o desempenho geral de uma função de hipótese h, fornecemos um conjunto de exemplos distintos dos usados para o treinamento e o chamamos de conjunto de testes. Dessa forma, não usamos os mesmos dados de treinamento para testar o algoritmo, pois teríamos um algoritmo viciado em nosso conjunto de dados. Essa divisão, geralmente é feita da seguinte maneira: 70% do conjunto de dados é dividido em um conjunto de treinamento e os 30% restantes são usados em um conjunto de testes [31].

A desvantagem dessa abordagem é que essa divisão implica que 30% dos dados serão deixados de lado pelo algoritmo. Isso pode implicar na perda significante de assertividade do modelo de predição. Na prática, os métodos de separação entre treinamento e testes mais usados são $60:40$, $70:30$ e $80:20$ e para conjuntos de dados muito grandes é $90:10$. Portanto, para minimizar a perda de dados no conjunto de treinamentos, pode-se usar uma técnica chamada validação cruzada [24].

O método *k-fold*, ou validação cruzada, é uma forma de melhorar o aprendizado do algoritmo validando a generalidade do modelo. O conjunto de dados de treinamento (*dataset*) é dividido em subconjuntos k e o algoritmo de aprendizado é repetido k vezes. Sendo assim, repetidamente um dos subconjuntos k é usado como o conjunto de testes e os outros subconjuntos k-*1* são utilizados pelo algoritmo como conjuntos de treinamento, como pode ser observado na Figura 2.3. Em seguida, a função de erro médio em todos os k ensaios é calculada [29]. Por fim, deve-se calcular a média E dos erros médios de cada um dos ensaios.

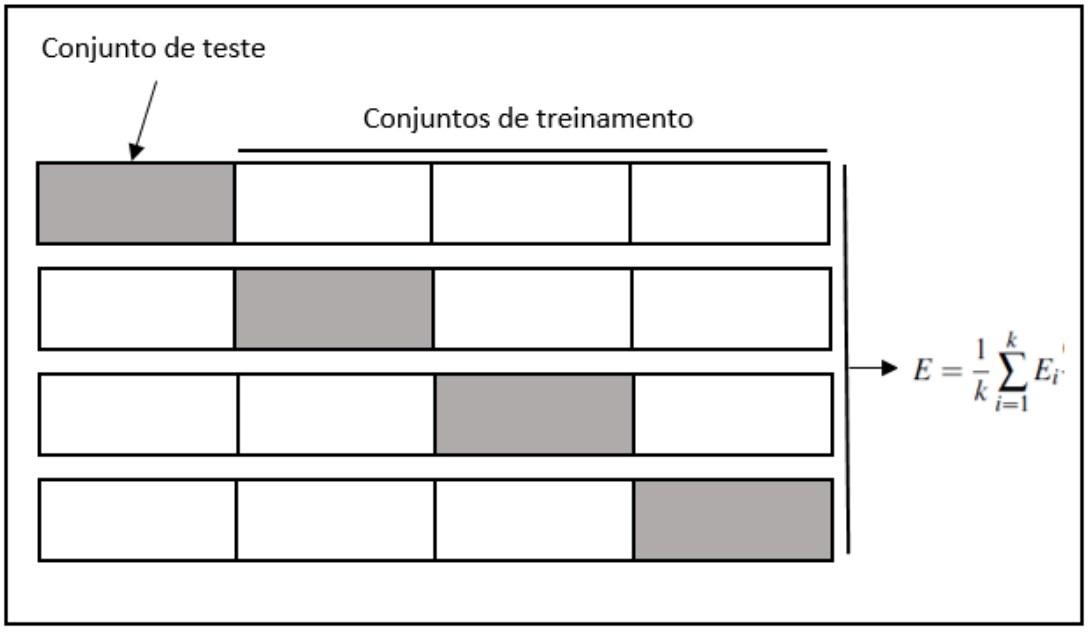

Figura 2.3: Conjunto de dados de treinamento usando validação cruzada *k-fold*.

Por exemplo, supondo que um conjunto de dados foi dividido em *4-fold* e o erro médio de cada ensaio foi de {0.93, 0.91, 0.93, 0.93}. Para calcular o desempenho geral do modelo temos que dividir a soma dos resultados pelo número de ensaios

$$\frac{0,93 + 0,91 + 0,93 + 0,93}{4} = 92,5\% \tag{2.6}$$

A vantagem deste método é que ele se importa menos em como os dados são divididos, fazendo com que todos os dados façam parte do treinamento e dos testes exatamente uma vez. Todos os dados ficam em um conjunto de testes exatamente uma vez, e em um conjunto de treinamento *k-1* vezes. A variância da estimativa resultante é reduzida quando *k* é aumentado. A desvantagem deste método é que o algoritmo de treinamento deve ser atualizado a partir do zero *k* vezes, o que significa que leva *k* vezes mais processamento para fazer uma avaliação.

2.3 Algoritmos de aprendizado de máquina supervisionado

2.3.1 Random Forest Classifier

Uma árvore de decisão é a decomposição de um problema muito complexo em subproblemas menos complexos. A ideia é a utilização da divisão para a conquista e, assim, de forma recursiva a mesma técnica é aplicada nos subproblemas. Tal capacidade de discriminar a árvore consiste em divisão em subespaços com a utilização de atributos e a cada subespaço se faz uma associação de uma classe. Na Figura 2.4 exibe-se um modelo simples árvore de decisão [32].

Na Figura 2.4, são exibidos os passos de uma árvore de decisão, em que os "nós"contêm um determinado atributo, nos quais os ramos descendentes são executados baseados nos valores de atributos, e cada folha está relacionada a uma classe, e o caminho a ser percorrido desde a raiz até a folha, está relacionada com uma regra de classificação [32].

O algoritmo Random Forest Classifier cria várias subamostras de árvores de decisão a partir do subconjunto do conjunto de treinamento selecionado aleatoriamente. Em seguida, agrega os votos de diferentes árvores de decisão para decidir a classe final do objeto de teste usando a média para melhorar a precisão preditiva e controlar o ajuste excessivo [33].

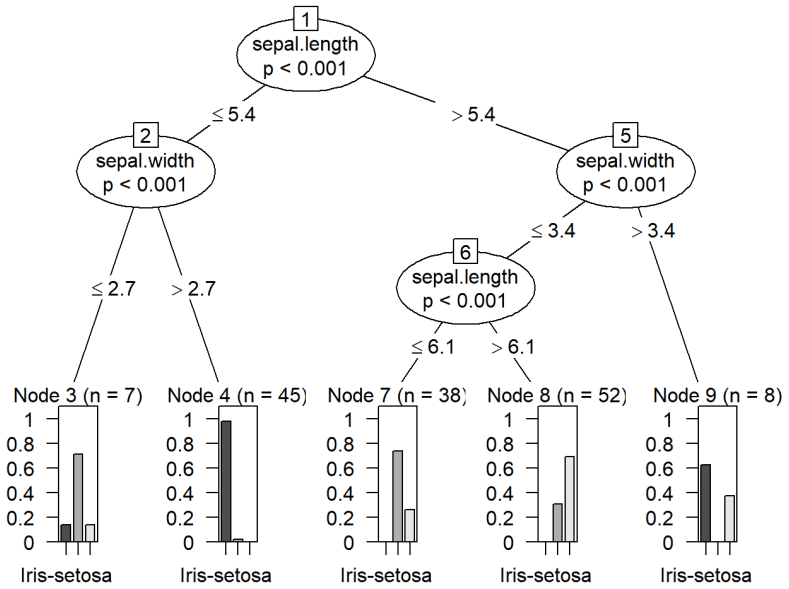

Figura 2.4: Modelo básico de uma árvore de decisão aplicada no conjunto de dados iris.

2.3.2 Naive Bayes

Naive Bayes é um classificador linear fácil de implementar, computacionalmente eficiente e tende a ter um desempenho particularmente bom em conjuntos de dados relativamente pequenos quando comparado com outros algoritmos [24]. O modelo probabilístico do Naive Bayes é baseado no teorema de Bayes e baseia-se na suposição de que os recursos em um conjunto de dados são mutuamente independentes. Na prática, a suposição de independência é frequentemente violada, mas o classificador Naive de Bayes ainda tende a ter um desempenho muito bom sob essa hipótese irrealista [34].

O teorema de *Bayes* que também é conhecido como lei de *Bayes* ou a regra de *Bayes*, é baseado no prévio conhecimento que descreve a probabilidade de um evento ocorrer. Ele possibilita alteração nas probabilidades com o objetivo de gerar novas evidências na obtenção de probabilidades posteriores [35].

Para entender como o Naive Bayes trabalha, precisa-se recapitular brevemente o conceito da regra de Bayes. O modelo de probabilidade que foi formulado por Thomas Bayes (1701-1761) é bastante simples, mas poderoso. Ele pode ser escrito em palavras simples como mostra a equação 2.7.

$$\text{probabilidade posterior} = \frac{\text{probabilidade condicional} \quad \times \quad \text{probabilidade anterior}}{\text{evidência}} \tag{2.7}$$

O teorema de Bayes constitui o núcleo de todo o conceito do algoritmos Naive Bayes. A probabilidade posterior, no contexto de um problema de classificação, pode ser interpretada como: “Qual é a probabilidade de um objeto

em particular pertencer à classe i conhecendo-se seus valores de características?".

Dessa forma, sendo:

- x_i o conjunto de dados de amostra i, $i \in \{1, 2, ..., n\}$,
- w_j a notação da classe j, $j \in \{1, 2, ..., m\}$,
- e $P(x_i \mid w_j)$ é a probabilidade de observar a amostra x_i, dado que pertence à classe w_j.

Portanto, a notação geral da probabilidade posterior pode ser escrita como na Equação 2.8 [22].

$$P(x_i \mid w_j) = \frac{P(x_i \mid w_j) \times P(w_j)}{P(x_i)} \tag{2.8}$$

2.3.3 K-Nearest Neighbors

O algoritmo K-Nearest Neighbors (k-NN) é um método não paramétrico usado para classificação e regressão [36]. A estatística não paramétrica (às vezes chamado de teste livre de distribuição) não pressupõe nada sobre a distribuição dos dados, ou seja, sabe-se apenas que os dados da população não têm uma distribuição normal [37]. Tanto na regressão como na classificação, a entrada consiste nas *k* amostras do conjunto de dados, como mostrado na Figura 2.5.

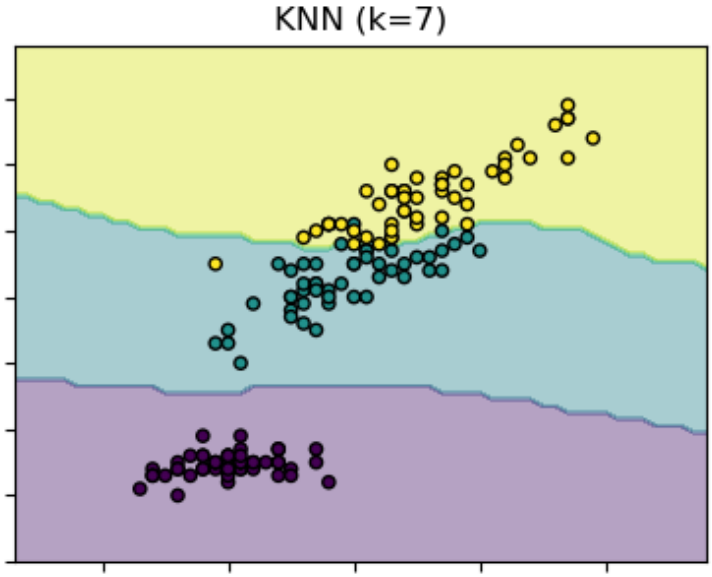

Figura 2.5: Mapeamento do conjunto de dados Iris [23] usando K-Nearest Neighbors [38].

Na classificação, o resultado é uma associação de classe baseado na similaridade a outros de mesma classe. Sendo assim, um objeto é classificado por

maioria de seus vizinhos, sendo o objeto atribuído à classe mais comum entre seus vizinhos mais próximos. Dessa forma, *k* é um número inteiro positivo, geralmente pequeno, e seu valor indica o número de vizinhos mais próximos envolvidos na determinação do rótulo de predição da classe nos dados do teste. Se $k = 1$, o objeto é simplesmente atribuído à classe desse vizinho mais próximo. Na regressão k-NN, a saída é o valor da propriedade para o objeto. Esse valor é a média dos valores dos seus vizinhos mais próximos.

O algoritmo k-NN é um tipo de aprendizagem baseada em instância, ou aprendizado preguiçoso, em que a função é apenas aproximada localmente e toda a computação é diferida até a classificação. Dessa forma, o algoritmo não aprende uma função discriminatória a partir dos dados de treinamento, mas, em vez disso, memoriza o conjunto de dados de treinamento.

Portanto, tanto para classificação quanto para regressão, uma técnica útil pode ser atribuir peso às contribuições dos vizinhos, de modo que os vizinhos mais próximos contribuam mais para a média do que os mais distantes. Por exemplo, um esquema de ponderação comum consiste em fornecer a cada vizinho um peso de $1/d$, em que *d* é a distância para o vizinho.

Os vizinhos são retirados de um conjunto de objetos para os quais a classe (para a classificação k-NN) ou o valor da propriedade do objeto (para a regressão k-NN) é conhecida. Isso pode ser pensado como o conjunto de treinamento para o algoritmo, embora não seja necessário um passo de treinamento explícito [39].

2.3.4 Support Vector Machines

Uma das abordagens mais influentes para a aprendizagem supervisionada é *Support Vector Machines* (SVM). Este modelo é conduzido por uma função linear $w^T x + b$, em que T um conjunto de treinamento e $w.x$ é o produto escalar entre os vetores w e x. Ao contrário de algoritmos de regressão, o algoritmo SVM não fornece probabilidades, mas apenas produz uma identidade de classe. O SVM prevê que a classe positiva está presente quando $w^T x + b$ é positivo. Do mesmo modo, prevê que a classe negativa esteja presente quando $w^T x + b$ é negativo [29].

Na prática, os dados de treinamento são transformados em um espaço de recursos dimensional separando os dados por meio de hiperplanos. Nesse caso, um modelo SVM linear é treinado para classificar os dados neste novo espaço de recursos, como mostrado na Figura 2.6. Então, a mesma função de mapeamento é usada para transformar dados novos e não vistos. No entanto, essa abordagem linear tem um problema de mapeamento quando se trata da classificação multicasse. Outo problema encontrado é que a construção dos novos recursos custa muito caro em termos computacionais, principalmente

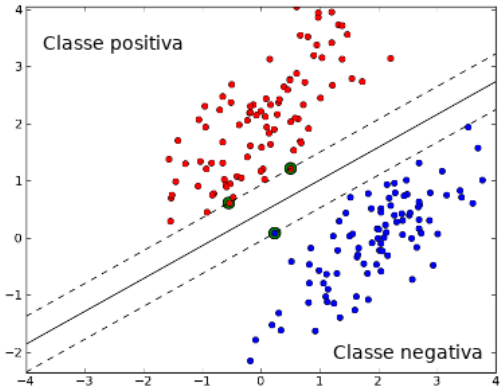

Figura 2.6: Exemplo do uso do algoritmo SVM linear.

para um conjunto de dados muito grande.

Uma inovação chave associada a máquinas de vetor de suporte é o truque do *kernel*. O truque do *kernel* consiste em observar que muitos algoritmos de aprendizagem de máquinas podem ser escritos exclusivamente em termos de produtos ponto ($\phi(\cdot)$) [40]. O operador $\cdot$ representa um produto interno análogo a $\phi(x)^T\phi(x^{(i)})$. Por exemplo, pode-se mostrar que a função linear usada pela máquina de vetor de suporte pode ser reescrita usando um modelo de *kernel* Gaussiano.

Resumidamente, devido ao termo exponencial, o valor de similaridade resultante cairá em um intervalo entre 1 (para amostras exatamente semelhantes) e 0 (para amostras muito diferentes) [24]. Dessa forma, é possível então encontrar hiperplanos não lineares capazes de separar esses dados. A Figura 2.7 exemplifica o uso prático do SVM usando *kernel* em um conjunto de dados.

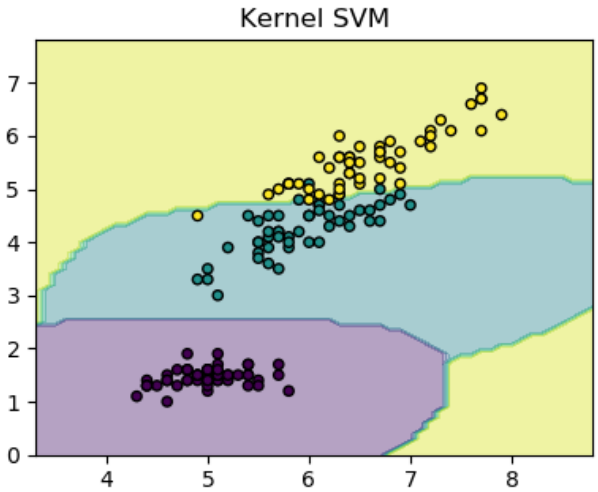

Figura 2.7: Mapeamento do conjunto de dados Iris [23] usando Support Vector Machines [38].

2.4 Ferramentas utilizadas

2.4.1 Scikit-Learn

Existe, em Python[1], uma biblioteca gratuita e de código aberto, chamada *scikit-learn* [42], que oferece uma grande variedade de algoritmos de aprendizagem de máquina. Essa biblioteca possui uma API (*Application Program Interface*) com uma interface amigável e de fácil uso que já está altamente otimizada para diferentes algoritmos de aprendizado de máquina. Além de otimizações, a biblioteca possui uma quantidade considerável de funções que ajudam a preprocessar e adequar os dados a serem trabalhados.

O *scikit-learn* surgiu, em meados de 2007, como com um projeto de David Cournapeau no Google Summer of Code. Posteriormente, no mesmo ano, Matthieu Brucher assumiu os trabalhos neste projeto como parte de sua tese de doutorado. O primeiro lançamento público dessa biblioteca foi feito em 2010 por funcionários do INRIA[2] que assumiram a liderança do projeto. Atualmente, quem assumiu o desenvolvimento dessa biblioteca são os membros da comunidade internacional de desenvolvimento [44].

Está sendo distribuído sobre a licença BSD (*Berkeley Software Distribution*) que estabelece poucas restrições para o uso do software. Vale a pena ressaltar, que essa licença é de código aberto e permite o uso comercial da biblioteca.

2.5 Trabalhos relacionados

Um trabalho já existente, que deu origem a essa proposta, é o do Fernando Maia da Mota [45]. Em sua abordagem, ele fez um processo de extração, transformação e carregamento dos dados que foram fornecidos pela Associação Sul-matogrossense de Produtores de Novilho Precoce e pela Embrapa Gado de Corte. Logo em seguida, construiu-se um modelo multidimensional para armazenar o resultado da primeira etapa. A terceira etapa é composta pela visualização e exploração dos dados armazenados. Por fim, realizou-se a aplicação de algoritmos de mineração de dados com a finalidade de descobrir padrões e relacionamentos vinculados ao grau de acabamento de gordura e ao rendimento de carcaça bovina.

Entretanto, algumas limitações nos dados fornecidos pela associação foram identificadas. Um bom exemplo disso, foi que os dados do armazenamento do peso vivo dos animais eram feitos na forma de média por lote. Ocasionando,

[1] Python é uma linguagem de programação interpretada, orientada a objetos e de alto nível com semântica dinâmica [41].

[2] Organização pública da França de cunho científico e tecnológico que existe desde janeiro de 1967 e tem como seu principal objetivo reunir pesquisadores e incentivar a pesquisa nas áreas de informática e automação [43].

assim, uma dificuldade para alcançar uma boa precisão em seus algoritmos de mineração de dados. Dessa forma, foi preciso utilizar novos dados, fornecidos pela Embrapa Gado de Corte, que foi considerado ideal para o seu trabalho por conter os dados dos abates por animais e não por lotes. Utilizando esses novos dados foi possível obter resultados com taxa de precisão entre 70% a 80%.

Portanto, os resultados de seu trabalho mostram que é possível utilizar técnicas de mineração de dados desde que o peso de cada animal seja conhecido durante o seu abate. Visto que, no contexto de seu trabalho, a classificação quanto ao rendimento de carcaça era um objetivo relevante. Em sua seção de "trabalhos futuros", Mota evidencia que é possível obter resultados mais assertivos se for utilizado um conjunto de dados que possua informações do rebanho abatido "porteira a dentro", ou seja, dados que informem como esse animal foi criado antes de ser finalizado.

Dentre os trabalhos encontrados, a maior parte deles aplicava o Support Machine Vector (SVM). Mesmo sendo computacionalmente caro, mostra-se útil em vários tipos de aplicações práticas [46]. Alguns autores obtiveram um resultado promissor para um conjunto de dados grande usando um modelo de predição multi-classe com SVM distribuído [47], em que usaram uma aproximação um-contra-um. A abordagem um-contra-um consiste em combinar cada classe a todas as outras classes.

Existe um estudo de comparação entre nove algoritmos SVM [48], usando validação cruzada para chegar a conclusão que um modelo baseado no algoritmo proposto por Weston & Watkins [49] pode dar melhor resultado em termos de precisão e velocidade.

Um outro artigo chega a usar, em conjunto multiclasse, os algoritmos SVM e K-NN para classificar caracteres tailandeses. O conjunto de dados possui mais de 60 características que é a somatória dos caracteres do alfabeto tailandês com suas pontuações e acentuações [50].

O algoritmo de predição K-NN também é usado em outro artigo para uma classificação multiclasse da qualidade da carne. O artigo mostra que essa análise é possível usando os parâmetros das cores encontradas em uma amostra de carne. Mesmo com o conjunto de dados sendo um pouco diferente do apresentado neste trabalho, com apenas quatro características numéricas, os resultados usando este algoritmo alcançaram um nível de precisão de 72% [51].

O trabalho de Chaturvedi *et al.* [52] assemelha-se com o tipo de conjunto de dados que este trabalho usa e utiliza Naive Bayes, SVM e Árvores de Decisão para predizer a severidade de *bugs*. Os autores mediram o desempenho de diferentes técnicas de aprendizado de máquina considerando o valor de precisão em todos os níveis de severidade e o número de melhores casos com diferentes

níveis de precisão. Um último exemplo, que usa SVM, Naive Bayes e K-NN, é um estudo que também faz a priorização de *bugs* reportados e compara os algoritmos usando validação cruzada. Esse estudo conta com um conjunto de dados grande, com mais de 50 características e com dados categóricos [53].

CAPÍTULO

3

Materiais e métodos

3.1 Abordagem

Tendo em vista implementar um classificador do grau de gordura de carcaças, a abordagem utilizada segue o esquema mostrado na Figura 3.1. O primeiro passo foi a obtenção dos dados de abate dos bovinos e seus respectivos processos produtivos.

Depois da obtenção do conjunto de dados, viu-se a necessidade de pré-processamento para remover colunas desnecessárias, amostras com valores insuficientes e seleção de características.

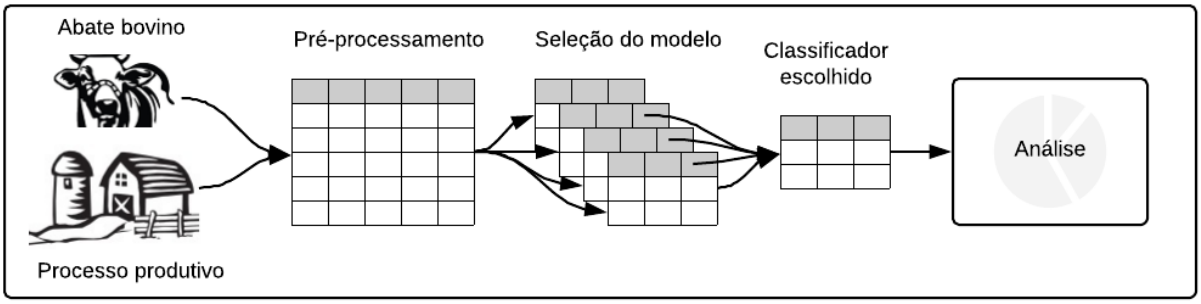

Figura 3.1: Fluxograma completo da metodologia utilizada.

Com os dados pré-processados, deu-se início a um processo de comparação dos algoritmos de aprendizado de máquina candidatos. Sendo assim, o algoritmo que apresentou o melhor resultado foi escolhido para ser a base de criação de um modelo de classificação.

Com o modelo de classificação em mãos, a otimização dos parâmetros pôde ser realizada com o objetivo de obter um modelo mais assertivo. Além disso,

foi possível concluir quais são as características que mais contribuem para o grau de gordura de carcaça.

Nestas etapas, levou-se em consideração aspectos do processo produtivo, como o tipo de suplementação utilizada, o tipo de confinamento e os itens de sustentabilidade utilizados pelos estabelecimentos rurais de Mato Grosso do Sul. Além disso, os dados do bovino durante o abate, como a maturidade, o peso da carcaça e o sexo do animal, foram levados em consideração.

3.2 Aquisição dos dados

O processo de aquisição de dados deu-se por meio de uma solicitação dos dados de abate de bovinos cadastrados no programa estadual Precoce MS e do processo produtivo de cada estabelecimento rural correspondente. A solicitação foi enviada para a Superintendência de Gestão da Informação (SGI) com o auxilio da Embrapa Gado de Corte e da Secretaria de Estado de Meio Ambiente, Desenvolvimento Econômico, Produção e Agricultura Familiar (SEMAGRO), ambas de Mato Grosso do Sul.

Os dados dos estabelecimentos rurais participantes do programa estadual foram entregues, pela SGI, em dois conjuntos de dados diferentes. O primeiro conjunto de dados compreendia os dados cadastrais (Tabela 3.1) e seus respectivos processos produtivos (Tabela 3.2 e Tabela 3.3). Este conjunto de dados constou com 1.595 estabelecimentos rurais cadastrados.

Tabela 3.1: Cinco amostras aleatórias dos dados cadastrais de um estabelecimento rural.

property_id	city	state	classification
5159	PEDRO GOMES	MS	21%
1167	CAMAPUA	MS	21%
4960	CAMAPUA	MS	26%
4514	TERENOS	MS	30%
5371	MARACAJU	MS	26%

O processo produtivo de um estabelecimento rural é definido por meio de um questionário online. O preenchimento desses dados é feito por responsáveis técnicos, que devem ter formação como médico veterinário, engenheiro agrônomo ou zootecnista e são corresponsáveis por essas informações [10].

As perguntas do questionário dividem-se em dois grupos:

- **"perguntas que não classificam":** que não aumentam a porcentagem do retorno financeiro para o produtor (Tabela 3.2); e
- **"perguntas que classificam":** que aumentam a porcentagem do retorno financeiro para o produtor (Tabela 3.3)

Tabela 3.2: Perguntas que não classificam o processo produtivo de um estabelecimento rural, seus respectivos rótulos no conjunto de dados e seus possíveis valores.

Perguntas	**Rótulos**	**Possíveis respostas**
Existem outros incentivos?	other_incentives	"Sim","Não"
Fabrica ração?	makes_ration	"Sim","Não"
Pratica suplementação a campo?	field_supplementation	"Sim","Não"
Executa o rastreamento SISBOV?	sisbov	"Sim","Não"
Faz parte da Lista Trace?	trace_list	"Sim","Não"
A área do estabelecimento rural é destinada na sua totalidade à atividade do confinamento?	total_area_confinement	"Sim","Não"

Tabela 3.3: Perguntas que classificam o processo produtivo de um estabelecimento rural, seus respectivos rótulos no conjunto de dados e seus possíveis valores.

Perguntas	**Rótulos**	**Possíveis respostas**
Dispõe de um sistema de identificação individual de bovinos associado a um controle zootécnico e sanitário?	individual_identification	"Sim","Não"
Faz controle de pastejo que atende aos limites mínimos de altura para cada uma das forrageiras ou cultivares exploradas, tendo como parâmetro a régua de manejo instituída pela Empresa Brasileira de Pesquisa Agropecuária (Embrapa)?	grazing_control	"Sim","Não"
O Estabelecimento rural apresenta atestado de Programas de Controle de Qualidade (Boas Práticas Agropecuárias - BPA/BOVINOS ou qualquer outro programa com exigências similares ou superiores ao BPA)?	quality_programs	"Sim","Não"
O Estabelecimento rural está envolvido com alguma organização que utiliza-se de mecanismos similares a aliança mercadológica para a comercialização do seu produto?	involved_in_organization	"Sim","Não"
A área manejada apresenta boa cobertura vegetal, com baixa presença de invasoras e sem manchas de solo descoberto em, no mínimo, 80% da área total de pastagens (nativas ou cultivadas)?	area_80_vegetation_cover	"Sim","Não"
A área manejada apresenta sinais de erosão laminar ou em sulco igual ou superior a 20% da área total de pastagens (nativas ou cultivadas)?	area_20_erosion	"Sim","Não"
Pratica recuperação de pastagem?	pasture_recovery	"Fertirrigação", "ILP - Integração Lavoura-Pecuária", "ILPF - Integração Lavoura-Pecuária-Floresta", "IFP - Integração Pecuária-Floresta", "Nenhum"
Pratica semi-confinamento?	semi_confinement	"Sim","Não"
Pratica confinamento?	confinement	"Sim","Não"

O segundo conjunto de dados engloba todos os abates individuais de bovinos, no período de 09/02/2017 até 23/01/2019, com o acabamento da carcaça equivalente (Tabela 3.4). Este conjunto de dados consta com 1.107.689 animais abatidos.

Tabela 3.4: Cinco amostras aleatórias do conjunto de dados dos abates de bovinos.

property_id	typification	maturity	carcass_weight	date_slaughter	carcass_fatness_degree
1	Macho INTEIRO	Dente de leite	362.50	2017-10-02	Gordura Mediana - acima de 3 a até 6 mm de espessura
1473	Macho CASTRADO	Dois dentes	252.00	2017-04-26	Gordura Escassa - 1 a 3 mm de espessura
4312	Macho CASTRADO	Quatro dentes	338.00	2017-05-08	Gordura Escassa - 1 a 3 mm de espessura
5068	Fêmea	Dois dentes	188.20	2018-01-03	Gordura Mediana - acima de 3 a até 6 mm de espessura
4452	Macho CASTRADO	Dente de leite	338.00	2018-05-15	Gordura Mediana - acima de 3 a até 6 mm de espessura

Cada amostra, da Tabela 3.4, representa o abate individual de bovinos. Ao abater um animal, o frigorífico registra a **tipificação** (Macho INTEIRO, Macho CASTRADO ou Fêmea), a **maturidade** (Dente de leite, Dois dentes, Quatro dentes, Seis dentes ou Oito dentes), o **peso da carcaça** (em kg), a **data do abate** e o **grau de gordura da carcaça** (Magra - Gordura ausente, Gordura Escassa - 1 a 3 mm de espessura, Gordura Mediana - acima de 3 a até 6 mm de espessura, Gordura Uniforme - acima de 6 e até 10 mm de espessura ou Gordura Excessiva - acima de 10 mm de espessura).

3.3 Pré-Processamento

Cada um dos conjuntos de dados contêm um identificador chamado *property_id* que liga cada abate ao seu respectivo estabelecimento rural de origem. Dessa forma, foi possível fazer a união dos dois mantendo uma linha para cada abate e o processo produtivo do bovino naquele momento.

O conjunto de dados resultante passou a ter apenas 1.056.586 amostras. Isso aconteceu devido a diferença de tempo na geração dos conjuntos de dados pela SGI. No total, 51.103 amostras de abates não possuíam identificadores de estabelecimentos rurais relacionados ao conjunto de dados do processo produtivo.

O próximo passo foi remover colunas que representam identificadores de banco de dados, dados cadastrais dos frigoríficos e colunas cujo valores são obtidos após a classificação do grau de gordura da carcaça. Visto que essas características não contribuem para o processo de aprendizado de máquina [54].

Escutar um especialista de negócio é uma das formas de eliminar ou criar novas características no conjunto de dados [55]. De acordo com o especialista em zootecnia, os dados relacionados a micro e mesorregião, que não estão no conjunto de dados, podem ser relevantes para saber o quão condições edafoclimáticas e sócio-econômicas de uma determinada região influenciam no grau de acabamento da carcaça. Para tal, usou-se os dados do município do estabelecimento rural para inferir duas novas características para esses valores. Levou-se em consideração a divisão interna do Estado de Mato Grosso do Sul [56].

Outra opinião relevante do especialista, foi em relação ao possível impacto

de determinada época do ano no grau de gordura resultante. Desta maneira, as datas de abate de um bovino foram usadas para inferir uma nova característica com a estação do ano na qual o animal foi terminado. De acordo com as datas de mudança de estações do Brasil [57].

O objetivo principal deste trabalho foi criar um classificador de grau de gordura da carcaça, levando em consideração um determinado animal e o processo produtivo que o gerou. Sendo assim, tem-se 5 classes ordinais que vão de Magra até Gordura Excessiva, como mostrado na Tabela 3.5.

Tabela 3.5: Classes para a classificação do grau de gordura da carcaça.

Rótulo	**carcass_fatness_degree**
Magra - Gordura ausente	1
Gordura Escassa - 1 a 3 mm de espessura	2
Gordura Mediana - acima de 3 e até 6 mm de espessura	3
Gordura Uniforme - acima de 6 e até 10 mm de espessura	4
Gordura Excessiva - acima de 10 mm de espessura	5

A maioria dos algoritmos de aprendizado de máquina requer que os valores das características sejam numéricos [58] para serem processados. Sendo assim, as colunas com valores categóricos devem ser transformadas em valores numéricos.

A coluna categórica *city* foi substituída por sua respectiva *latitude* e *longitude*. A coluna *state* foi removida por conter sempre o mesmo valor que representa o estado de Mato Grosso do Sul (MS). A coluna *maturity* teve seus valores substituídos pela quantidade de dentes definitivos em valores inteiros.

A **tipificação** teve seus valores codificados usando *label encoder* com a seguinte ordem para os valores categóricos “Macho INTEIRO”, “Macho CASTRADO” e “Fêmea” resultando nos valores inteiros 0, 1 e 2 respectivamente.

Todas as perguntas cujas possíveis respostas estavam restritas a “Não” ou “Sim” tiveram seus valores substituídos por 0 e 1 respectivamente. No caso das estações do ano verão, outono, inverno e primavera, elas tiveram seus valores substituídos por 0, 1, 2 e 3 respectivamente.

As onze microrregiões Alto Taquari, Aquidauana, Baixo Pantanal, Bodoquena, Campo Grande, Cassilândia, Dourados, Iguatemi, Nova Andradina, Paranaíba e Três Lagoas receberam os valores $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$. As quatro mesorregiões Centro Norte, Leste, Pantanal e Sudoeste receberam os valores $[0, 1, 2, 3]$.

Houve a substituição da data do abate (*date_slaughter*) pelo mês do abate. Segundo o especialista, o dado que referencia o mês é mais relevante do que a data do abate em si. A nova característica (*month_slaughter*) recebeu valores de 0 a 11 de acordo com a ordem anual em que o mês ocorre.

Na conversão de características nominais com mais de uma opção por amostra pode-se usar a técnica *one hot encoding* e convertê-las em novas ca-

racterísticas duplicadas, onde cada uma representa um dos valores da característica. Nessa técnica, a presença de um nível é representada por 1 e a sua ausência por 0 [24].

A técnica de *one hot encoding* foi utilizada na coluna *pasture_recovery* resultando na criação de quatro novas características *fertigation*, *fli*, *clfi* e *lfi*. Cada nova característica recebia o valor 0 quando não era aplicada naquele estabelecimento rural e o valor 1 quando era aplicada.

A unidade de medida usada para determinada característica pode afetar negativamente a análise dos dados. A técnica de normalização *min-max* [59] consiste em expressar os valores de uma característica em unidades de medida menores. Dessa forma, os dados são transformados em uma distribuição mais regular, tal como $[0,0;1,0]$.

Por fim, ao término do pré-processamento dos dados, tem-se o conjunto de dados resultante com 28 características para treinamento e 1 característica classificadora, como mostra a Tabela 3.6.

Tabela 3.6: Descrição do conjunto de dados completo após ser pré-processado.

Característica	**Média aritmética**	**Desvio padrão**	**Mínimo**	**Máximo**	**Média aritmética (normalizada)**	**Desvio padrão (normalizado)**
typification	0,8025	0,8585	0	2	0,6196	0,4482
maturity	2,0141	1,9251	0	8	0,4127	0,3404
carcass_weight	267,9293	45,8414	150	450	0,3871	0,1467
classification	24,4813	3,6171	21	30	0,3447	0,3749
other_incentives	0,0544	0,2268	0	1	0,0457	0,2067
makes_ration	0,6580	0,4743	0	1	0,6643	0,4699
total_area_confinement	0,0194	0,1380	0	1	0,0046	0,0677
area_80_vegetation_cover	0,8060	0,3953	0	1	0,7717	0,4195
area_20_erosion	0,0020	0,0456	0	1	0,0022	0,0473
individual_identification	0,7472	0,4345	0	1	0,7479	0,4306
sisbov	0,4821	0,4996	0	1	0,3830	0,4847
grazing_control	0,8162	0,3872	0	1	0,7764	0,4164
trace_list	0,4456	0,4970	0	1	0,3527	0,4761
quality_programs	0,0254	0,1576	0	1	0,0182	0,1307
involved_in_organization	0,3374	0,4728	0	1	0,3415	0,4728
confinement	0,6135	0,4869	0	1	0,5476	0,4958
semi_confinement	0,4995	0,5000	0	1	0,5552	0,4944
field_supplementation	0,9049	0,2932	0	1	0,9418	0,2290
fertigation	0,0933	0,2909	0	1	0,1114	0,3122
lfi	0,0495	0,2170	0	1	0,0509	0,2175
fli	0,4200	0,4935	0	1	0,3430	0,4729
clfi	0,0505	0,2191	0	1	0,0295	0,1664
latitude	-20,6525	1,5130	-23,9702	-17,5769	0,5288	0,2102
longitude	-54,1835	1,3038	-57,8825	-51,0936	0,5666	0,1891
month_slaughter	6,8078	3,3633	0	11	0,5807	0,2917
season_slaughter	1,9211	1,0919	0	3	0,6089	0,3535
microregion	5,1042	3,5738	0	10	0,5137	0,3702
mesoregion	1,2321	1,1499	0	3	0,3422	0,3533
carcass_fatness_degree	3,1264	1,5934	1	5	3,1264	1,5934

3.4 Pré-Análise

A Figura 3.2 mostra a distribuição das classes de acabamento no conjunto de dados. De imediato, percebe-se que o conjunto de dados é desbalanceado. O conjunto de dados continha $0,48\%$ de suas amostras destinadas a classe 1, $40,25\%$ a classe 2, $53,31\%$ a classe 3, $5,94\%$ a classe 4 e $0,02\%$ a classe 5.

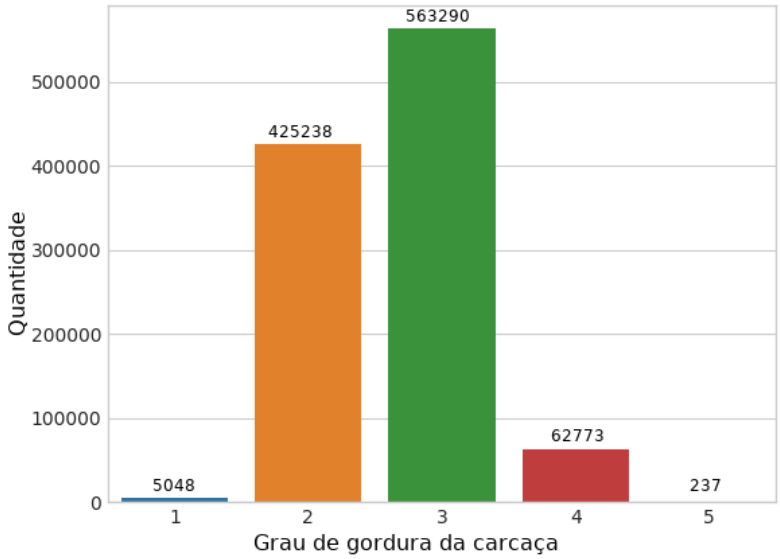

Figura 3.2: Distribuição do grau de gordura da carcaça.

Um conjunto de dados desbalanceado pode afetar negativamente a fase de aprendizado e, consequentemente, a classificação dos algoritmos de aprendizado de máquina [60]. Com a intenção de balancear os dados, existem dois métodos que se destacam: *over-sampling* que replica amostras da classe minoritária e *under-sampling* que elimina amostras da classe majoritária [61].

A técnica de *under-sampling* chamada Edited Nearest Neighbours (ENN) [62] foi utilizada. Essa técnica aplica um algoritmo de vizinhos mais próximos e remove as amostras que não concordam "o suficiente" com a sua vizinhança. Para cada amostra na classe a ser sub amostrada, os vizinhos mais próximos são calculados e, se o critério de seleção não for atendido, a amostra é removida.

Em contrapartida, a técnica de *over-sampling* chamada Synthetic Minority Over-sampling Technique (SMOTE) gera novas amostras "sintéticas" por interpolação operando direto no âmbito das características ao invés dos dados [63].

Contudo, o uso da técnica SMOTE pode gerar amostras ruidosas interpolando novos pontos entre valores marginais e valores isolados. Esse problema pode ser resolvido limpando o espaço resultante da super amostragem com ENN. Sendo assim, o uso das técnicas *over-sampling* SMOTE e *under-sampling* ENN combinadas, chamada SMOTEENN, tem gerado melhores resultados [64]. Isto posto, todas essas três técnicas de balanceamento terão seus resultados comparados nest trabalho.

A melhor maneira de saber o quão bem um modelo irá generalizar para novos casos é realmente experimentá-lo em novos casos [55]. Sendo assim, o primeiro passo foi dividir o conjunto de dados em dois: 80% para treino e 20% para testes. A divisão foi feita de forma aleatória e levando em consideração a porcentagem de cada classe.

O balanceamento foi feito no conjunto de dados de treinamento. O resultado, depois do balanceamento, pode ser observado na Figura 3.3. O conjunto de dados de testes será usado para validar cada um dos modelos testados. Dessa forma, o conjunto de dados de testes não foi alterado, permanecendo com 211.318 amostras.

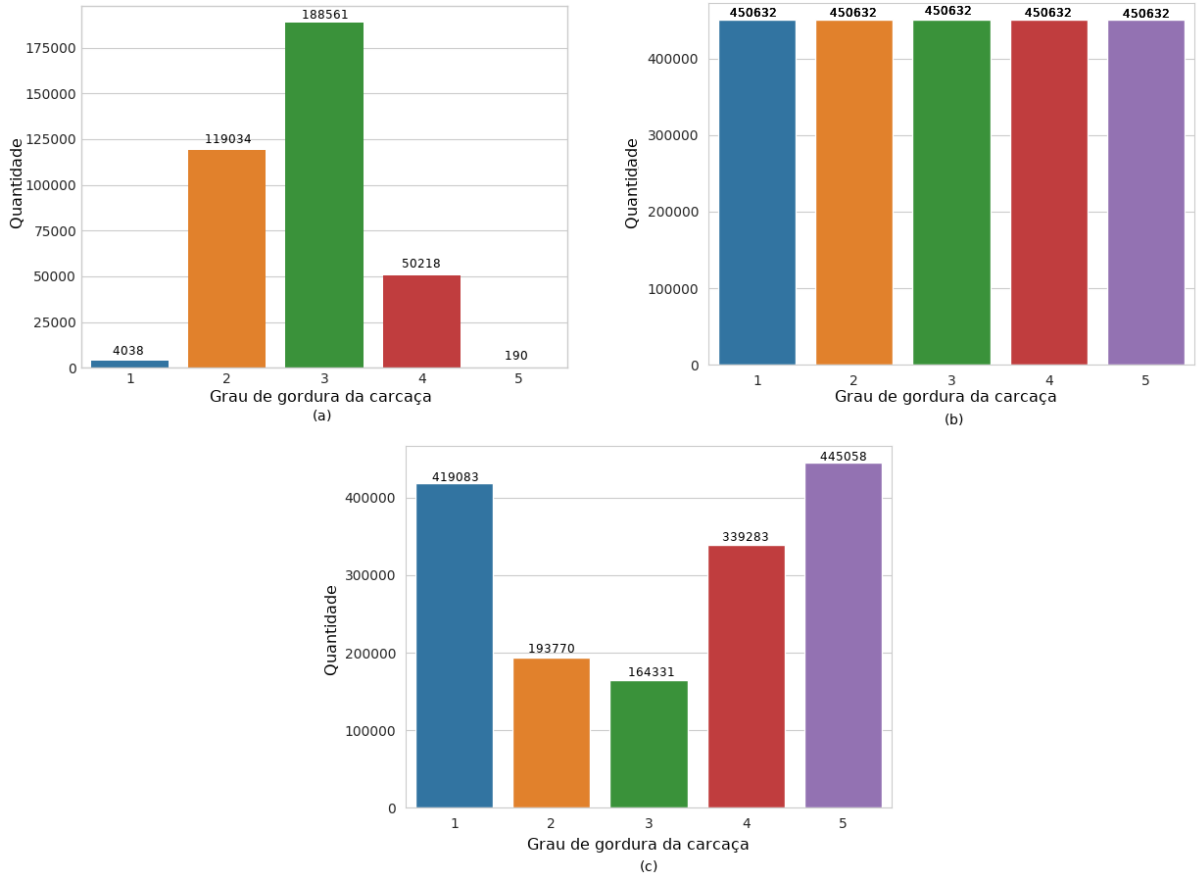

Figura 3.3: Comparação da distribuição das classes com balanceamento do conjunto de dados usando técnicas *under-sampling*, *over-sampling* e híbridas. (a) *under-sampling* com ENN; (b) *over-sampling* com SMOTE; e (c) combinação das técnicas SMOTE + ENN.

3.5 Métricas de desempenho

O desempenho da classificação de algoritmos de Aprendizado de Máquina pode ser avaliado usando medidas quantitativas [65]. Dentre as métricas mais usadas para avaliar modelos, destacam-se *matriz de confusão*, *precision*, *recall* e *f1 score*.

A matriz de confusão é, simplesmente, uma matriz que relata as contagens das previsões feitas por um classificador em relação ao conjunto de dados de teste. Sendo assim, em uma matriz de confusão os valores das linhas retratam as classes atuais e os das colunas indicam as classes previstas [66]. Um exemplo de matriz de confusão pode ser observado na Tabela 3.7, em

que os rótulos das colunas representam as classes e os valores entre elas representam as predições.

Tabela 3.7: Dois exemplos de matrizes de confusão: (a) Matriz de confusão com quatro classes; (b) Matriz de confusão com duas classes.

(a)

	A	**B**	**C**	**D**
A	137	13	3	0
B	1	55	1	0
C	2	4	84	0
D	3	0	1	153

(b)

	P	**N**
P	VP	FN
N	FP	VN

A forma mais comum de analisar uma matriz de confusão pode ser observada na Tabela 3.7 (b). Nesse exemplo, a matriz de confusão M possui duas classes, a classe P (positivo) e a classe N (negativo). O valores do índice M_{22} da matriz indicam os valores classificados corretamente como P, ou seja, verdadeiros positivos (VP). O índice M_{23} mostra os resultados que foram erroneamente classificados como não sendo P, ou seja, falsos negativos (FN). Já no índice M_{32} os valores apontam todos os resultados classificados erroneamente como P, ou seja, falsos positivos (FP). Por fim, os resultados do índice M_{33} mostram os valores classificados corretamente como não sendo P, ou seja, verdadeiros negativos (VN).

A fim de exemplificar os conceitos apresentados, a Tabela 3.8 mostra os valores de VP, FN, FP e VN para as classes A, B, C e D. Cada coluna representa uma classe da matriz de confusão da Tabela 3.7 (a) e cada linha representa os valores (destacados de cinza) de VP, FN, FP e VN para cada classe A, B, C e D.

Tabela 3.8: Exemplos de verdadeiros positivos, falsos negativos, falsos positivos e verdadeiros negativos para as classes A, B, C e D.

A

		A	**B**	**C**	**D**
VP	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
FN	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
FP	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
VN	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153

B

		A	**B**	**C**	**D**
VP	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
FN	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
FP	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
VN	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153

C

		A	**B**	**C**	**D**
VP	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
FN	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
FP	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
VN	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153

D

		A	**B**	**C**	**D**
VP	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
FN	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
FP	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153
VN	**A**	137	13	3	0
	B	1	55	1	0
	C	2	4	84	0
	D	3	0	1	153

A principal métrica utilizada para validar algoritmos de Aprendizado de Máquina é a *accuracy*, que é a fração de previsões de acertos do modelo testado [67]. A *accuracy* é calculada a partir do total de acertos (VP e VN) e dividida pelo conjunto total dos dados (VP, FN, FP e VN), como mostra a Equação 3.1.

$$Accuracy = \frac{(VP + VN)}{(VP + FN + FP + VN)} \tag{3.1}$$

A *accuracy*, por si só, pode levar a interpretações tendenciosas quando trabalha-se com um conjunto de dados com desequilíbrio de classes em que há uma disparidade significativa entre o número de rótulos positivos e negativos [68]. Dessa forma, métricas mais coerentes como *precision* (Equação 3.2) e *recall* (Equação 3.3) podem ser utilizadas para auxiliar a entender os resultados apresentados [24].

$$Precision = \frac{VP}{VP + FP} \tag{3.2}$$

$$Recall = \frac{VP}{VP + FN} \tag{3.3}$$

Precision é definida como o número de VP dividido pelo número de VP somados com o número de FP e representa a fração de amostras corretamente classificadas. *Recall* é a quantidade de VP dividida pela quantidade de VP somada a quantidade de FN e expressa a proporção de amostras positivas que são detectadas corretamente pelo classificador.

No geral, as medidas *precision* e *recall* são combinadas para medir o desempenho do modelo, gerando uma equação conhecida como *f1-score*, que é a média harmônica entre ambas as métricas. O cálculo de *f1-score* é mostrado na Equação 3.4.

$$f1\_score = \frac{(2 \times precision \times recall)}{(precision + recall)} \tag{3.4}$$

As matrizes de confusão dos resultados são apresentadas com os valores normalizados, ou seja, é mostrado o percentual de acerto e erro de cada classe. As cores mais fracas representam valores maiores e as cores mais fortes representam valores menores. Portanto, fica fácil visualizar onde existe confusão entre as classes e qual a porcentagem de acertos e erros de cada modelo testado.

3.6 Seleção do modelo

Nas próximas etapas do trabalho, usou-se uma biblioteca em Python, chamada *scikit-learn*, para construir, testar e validar diferentes classificadores.

Os modelos que foram comparados nesse trabalho são: Multinomial Naive Bayes (MNB) [69], Random Forest Classifier (RFC) [33], K-Nearest Neighbours (KNN) [39] e Support Vector Machines (SVM) [70]. Todos eles foram aplicados no mesmo conjunto de dados balanceado e as mesmas métricas foram utilizadas como resultado para validação.

Para validar os modelos, uma abordagem básica é a utilização da técnica chamada *cross-validation*. O conjunto de treinamento é dividido em k conjuntos menores, no qual o seguinte procedimento é executado para cada uma das k divisões:

1. Um modelo é treinado usando as $k - 1$ divisões como dados de treinamento;
2. O modelo resultante é validado na parte restante dos dados.

A medida de desempenho apontada pela validação cruzada é então a média dos valores de *accuracy* para cada *loop*. As métricas *matriz de confusão*, *precision*, *recall* e *f1 score* foram utilizadas no conjunto de testes para auxiliar a ter *insights* durante a validação do modelo final.

Em todos os modelos, aplicava-se a técnica *cross-validation* com *5-folds* preservando a porcentagem de amostras em cada classe. O índice de randomização utilizado para garantir que os conjuntos de dados de treinamento e teste sejam sempre os mesmos foi 42. Na comparação dos modelos utilizou-se como métrica principal a média da *accuracy* de todos os folds, por modelo.

Cada *divisão* da validação cruzada conta com um conjunto de treinamento com 80% dos dados e um conjunto de testes com 20%. O conjunto de treinamento é balanceado e o modelo é treinado, conforme mostra a Figura 3.4. Após o treinamento, o conjunto de teste é usado para validação e os resultados da classificação são coletados.

Os resultados de cada divisão são combinados para obter a matriz de confusão e calcular as métricas *precision*, *recall* e *f1 score*. Os resultados e discussões em torno da seleção do modelo serão apresentados no próximo capítulo, juntamente com o modelo escolhido e como ele foi melhorado para obter maior acurácia.

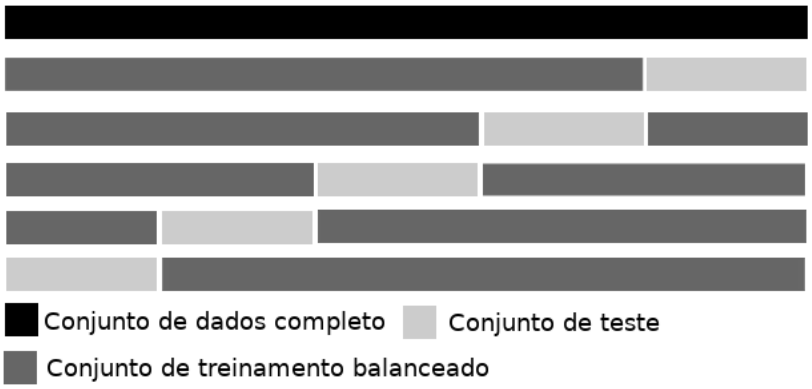

Figura 3.4: Visualização do processo de validação cruzada com o conjunto de treinamento balanceado.

CAPÍTULO

4

Resultados e discussões

Os resultados da comparação entre os modelos, utilizando os conjuntos de dados balanceado e desbalanceado, podem ser observados na Tabela 4.1.

Tabela 4.1: Comparação da média da *accuracy* entre os quatro modelos escolhidos e o desvio padrão obtido pelos resultados de cada dobra da validação cruzada. Siglas: Multinomial Naive Bayes (MNB), Random Forest Classifier (RFC), K-Nearest Neighbours (KNN) e Support Vector Machines (SVM)

	Accuracy			
Modelo	Desbalanceado	ENN	SMOTE	SMOTEENN
MNB	56,87% $(+/-)0,067$	57,15% $(+/-)0,102$	28,02% $(+/-)0,176$	13,13% $(+/-)0,343$
RFC	65,30% $(+/-)0,056$	66,61% $(+/-)0,075$	63,60% $(+/-)0,119$	65,61% $(+/-)0,089$
KNN	68,35% $(+/-)0,061$	68,04% $(+/-)0,048$	65,12% $(+/-)0,077$	64,81% $(+/-)0,127$
SVM	53,88% $(+/-)0,061$	56,11% $(+/-)0,116$	55,43% $(+/-)0,121$	57,82% $(+/-)0,337$

Os resultados obtiveram desvio padrão muito baixo, como pode ser observado na Tabela 4.1, para todas as dobras da validação cruzada aplicadas nos modelos. Isso mostra que os resultados para cada *fold* não estão viciados e os dados estão estratificados levando em consideração a porcentagem de cada classe.

Ao considerar os resultados obtidos, percebe-se que os modelos que generalizaram melhor foram os aplicados em dados balanceados usando ENN. Mais especificamente os algoritmos RFC e KNN. No entanto, analisar apenas a *accuracy* pode levar a uma estimativa otimista caso o classificador seja tendencioso [68].

Esse problema pode ser superado quando os resultados são analisados observando a *matriz de confusão* normalizada (Figura 4.1). Ao treinar os modelos, classificar usando o conjunto de dados de teste e imprimir suas respectivas matrizes de confusão, calculando as métricas *precision*, *recall* e *f1 score*,

obteve-se os resultados da Tabela 4.2.

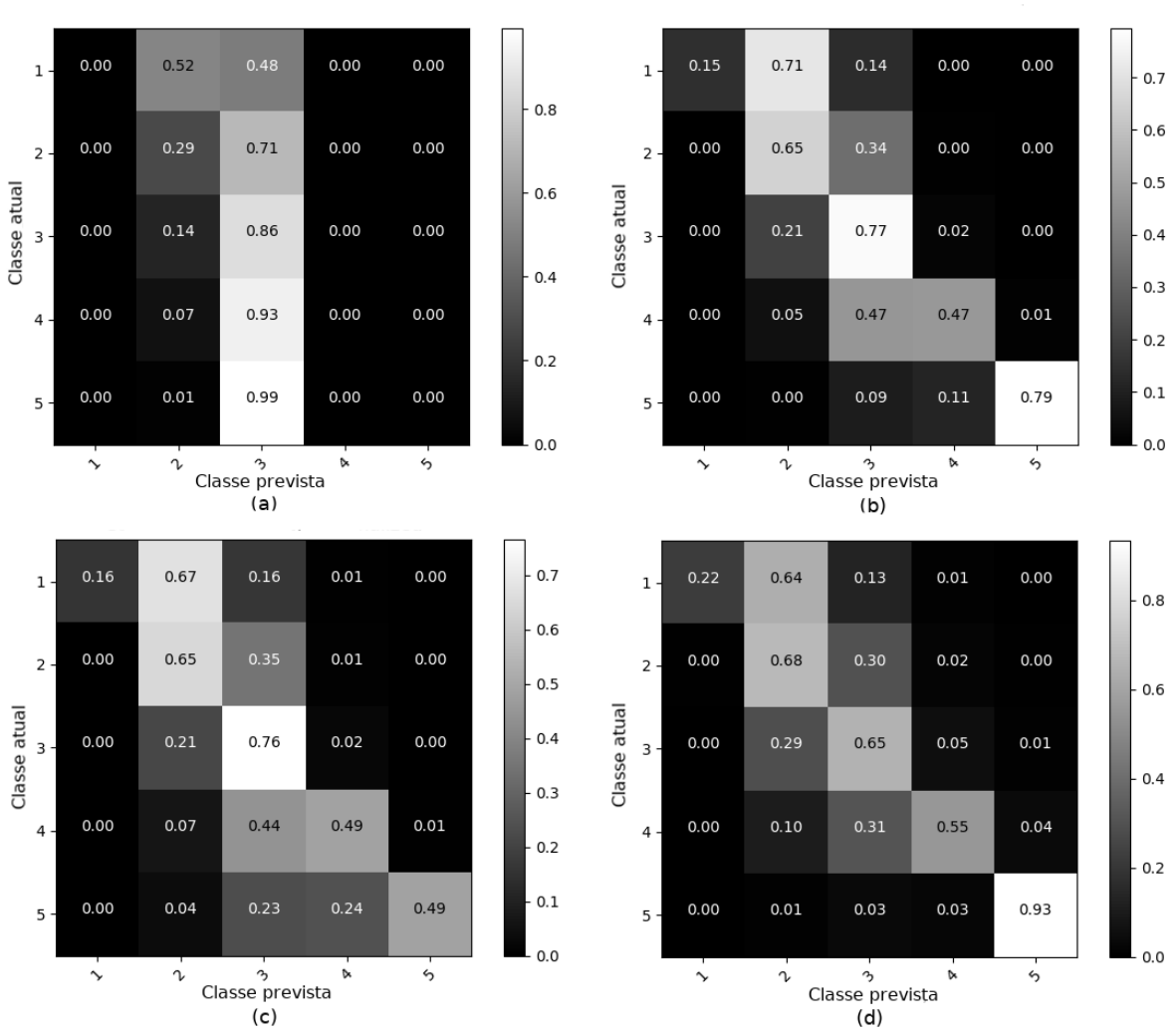

Figura 4.1: Resultados normalizados da predição dos algoritmos no conjunto de dados de testes, com a representação em escala de 0,00 a 1,00, para análise de erro: (a) matriz de confusão normalizada do algoritmo MNB; (b) matriz de confusão normalizada do algoritmo RFC; (c) matriz de confusão normalizada do algoritmo K-NN; e (d) matriz de confusão normalizada do algoritmo SVM.

Ao observar os resultados da classe 1 e 2 na *matriz de confusão* normalizada da Figura 4.1, pode-se constatar que o algoritmo SVM faz uma distinção melhor entre essas classes. O produtor rural recebe incentivo apenas para carcaças com grau de gordura 2, 3 e 4. Dessa forma, é importante que o número de falsos positivos seja baixo nos extremos da matriz, mais especificamente para as classes 1 e 5.

Aprendizado de máquina é o estudo de algoritmos que aumentam sua capacidade de descobrir padrões de forma automática de acordo com a experiência [26]. Dessa forma, a tarefa de descobrir padrões pode ser facilitada e consumir menos tempo removendo características que são irrelevantes ou redundantes em relação à tarefa a ser aprendida. Esse processo é chamado de seleção de características [71] [72].

Tabela 4.2: Comparação das métricas *Precision*, *Recall* e *F1 score* para cada uma das 5 classes, com suas respectivas médias, entre os modelos. O cálculo da média levou em consideração o peso de cada classe no conjunto de dados.

Modelo	**Métrica**	**1**	**2**	**3**	**4**	**5**	**Média**
MNB	Precision	00,00%	58,22%	56,96%	00,00%	00,00%	53,79%
	Recall	00,00%	28,52%	85,77%	00,00%	00,00%	57,20%
	F1 score	00,00%	38,28%	68,45%	00,00%	00,00%	51,90%
RFC	Precision	72,90%	69,23%	71,25%	71,74%	20,04%	70,47%
	Recall	14,60%	65,23%	77,48%	47,36%	79,32%	70,46%
	F1 score	24,33%	67,17%	74,24%	57,06%	32,00%	70,12%
KNN	Precision	65,28%	68,36%	70,97%	66,30%	15,39%	69,60%
	Recall	15,83%	64,60%	76,44%	48,57%	48,52%	69,73%
	F1 score	25,48%	66,43%	73,61%	56,07%	23,37%	69,43%
SVM	Precision	31,90%	62,66%	71,41%	47,48%	02,37%	66,26%
	Recall	22,33%	67,55%	64,98%	55,02%	93,25%	65,22%
	F1 score	26,27%	65,01%	68,04%	50,97%	04,62%	65,60%

A abordagem utilizada nesse trabalho é um método *wrapper* [73] conhecido como *recursive feature elimination* (RFE) [74] [75]. Métodos *wrapper* fazem uso do modelo de classificação para medir a importância do conjunto de características. Portanto, as características selecionadas dependem do modelo de classificação usado [76]. Por isso, a seleção de características foi feita depois da seleção do modelo.

O classificador RFC foi utilizado, junto com RFE, para fazer a seleção de características. Segundo o classificador, as características mais relevantes para o modelo são apresentadas na Figura 4.2. Todas as características desse conjunto de dados foram pontuadas com algum valor que representa sua importância no conjunto de dados completo em uma escala de 0 a 1. Logo, esse resultado foi de encontro do objetivo específico 2 desse trabalho.

A característica mais importante *carcass_weight* recebeu o maior score, $0,3503$, e a característica *area_20_erosion* recebeu o menor valor, $0,0003$. Esse cálculo é feito no momento da construção da árvore de decisão pelo algoritmo RFC, onde as características mais relevantes são os nós mais próximos da raiz e as com menor importância são folhas.

Ao aplicar a técnica RFE as características *fertigation*, *lfi*, *other_incentives*, *total_area_confinement*, *area_20_erosion* e *quality_programs* foram removidas do conjunto de dados. A remoção dessas características pode ser atribuído ao fato da alta sensibilidade das mesmas. Para tal, pode-se observar que na Tabela 3.6, mais especificamente a coluna Média aritmética, os valores dessas características variam entre 0 ou 1 e, em mais de 95% das amostras, tendem a ser 0. Por fim, o subconjunto final passou a ter 22 características base e 1 que representa a classe alvo.

Outra forma de melhorar as métricas é encontrar os melhores hiperparâ-

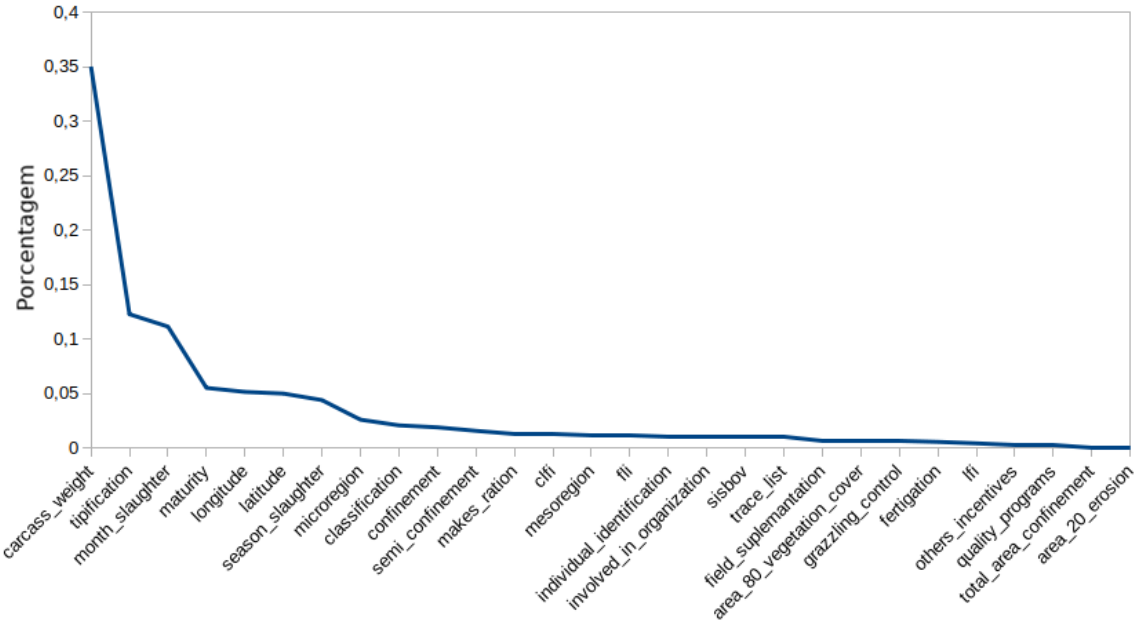

Figura 4.2: Características mais relevantes de acordo com o algoritmo RFC e a técnica de seleção de características RFE. O eixo y mostra a porcentagem de importância que a característica representa no conjunto.

metros para um algoritmo em determinado conjunto de dados. O processo de seleção empírica dos melhores hiperparâmetros para um modelo pode demorar muito e resultar em uma *accuracy* inferior [77]. Uma das técnicas mais utilizadas para seleção de hiperparâmetros de forma automática é *grid search* com validação cruzada [58], que é um processo de busca exaustiva sobre um conjunto de dados específico.

Na Tabela 4.3 pode-se observar a coluna “Vetor de parâmetros"que mostra os possíveis valores para cada hiperparâmetro de cada modelo de classificação testado. A técnica *grid search* vai usar esse vetor para fazer a busca exaustiva e analisar todas as possíves combinações. Todas as comparações são feitas usando validação cruzada.

Tabela 4.3: Comparação da média da *acurácia* e *f1 score* entre os quatro modelos escolhidos após a otimização com *grid search*.

Modelo	Vetor de parâmetros	Melhores Parâmetros	Acurácia	F1 score
MNB	$alpha = [1, 0.1, 0.01, 0.001, 0.0001, 0.00001]$	$alpha = 0.01$	57,72% $(+/-)0,081$	51,87%
RFC	$n\_estimators = [100, 250, 500, 750]$ $min\_samples\_leaf = [1, 2, 3, 5, 10, 20]$ $min\_samples\_split = [2, 4, 6, 8, 10, 20]$ $max\_features = [“sqrt”, “log2”, None]$ $class\_weight = [“balanced”]$ $max\_depth = [25, 50, 75]$	$n\_estimators = 250$ $min\_samples\_leaf = 1$ $min\_samples\_split = 6$ $max\_features = “sqrt”$ $class\_weight = “balanced”$ $max\_depth = 50$	70,45% $(+/-)0,139$	70,14%
KNN	$weights = [“uniform”, “distance”]$ $n\_neighbors = [1, 2, 3, 4, 5, 10, 15, 20]$	$weights = [“distance”]$ $n\_neighbors = [2]$	69,90% $(+/-)0,098$	69,65%
SVM	$C = [0.125, 0.5, 2, 8, 16, 64, 128, 256, 512]$ $gamma = [0.0625, 0.125, 0.5, 1, 2, 8, 16]$ $kernel = [“rbf”, “poly”]$	$C = 128$ $gamma = 8$ $kernel = “rbf”$	70,11% $(+/-)0,003$	69,89%

Depois de analisados, os candidatos são ranqueados pelo resultado de sua

accuracy e o candidato com maior resultado é apresentado como o melhor modelo. As *matrizes de confusão* normalizadas da Figura 4.3 mostram os resultados da predição do conjunto de dados de teste utilizando os melhores hiperparâmetros.

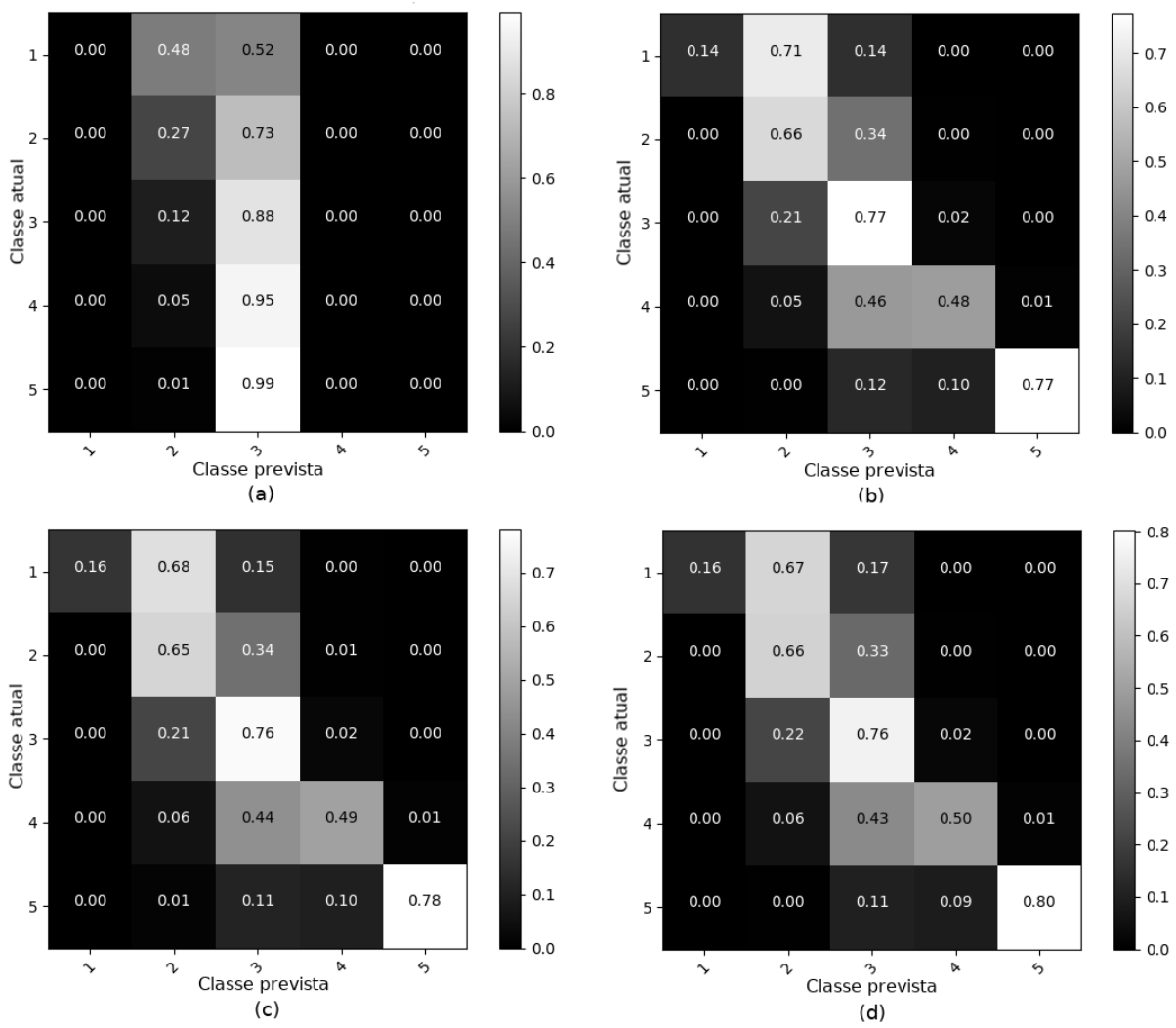

Figura 4.3: Resultados do algoritmo RFC quando executado sobre o conjunto de testes. (a) Matriz de confusão; (b) Matriz de confusão normalizada.

Ao observar a matriz de confusão normalizada (Figura 4.3), nota-se um aumento significante na porcentagem de acertos do algoritmo SVM, principalmente nas classes 3, 4 e 5. A Tabela 4.3 confirma esse aumento com o aumento nos valores da métrica *f1 score* e *accuracy*. O uso da técnica *grid search* auxiliou a construir modelos mais estáveis. Dessa forma, pode-se dar início as análises estatísticas para descobrir qual dos algoritmos testados servirá como modelo final.

Para verificar se os classificadores testados diferem estatisticamente em relação ao desempenho, utilizou-se a análise de variância unidirecional ANOVA [78]. Por meio do teste ANOVA chegou-se ao valor-p de *0,0436* que indica uma diferença estatisticamente significativa no desempenho médio das

técnicas testadas a um nível de significância de 5% usando *f1 score* como métrica.

O teste *post-hoc* de Tukey [79] foi aplicado, para comparar as médias que diferem estatisticamente com o limite de significância definido em 5%. O resultado mostrou que os classificadores SVM e RFC são semelhantes e não existe diferença estatística significativa de desempenho entre os dois classificadores. Sendo assim, concluem-se os testes dos algoritmos de aprendizado de máquina de acordo com o objetivo específico 1.

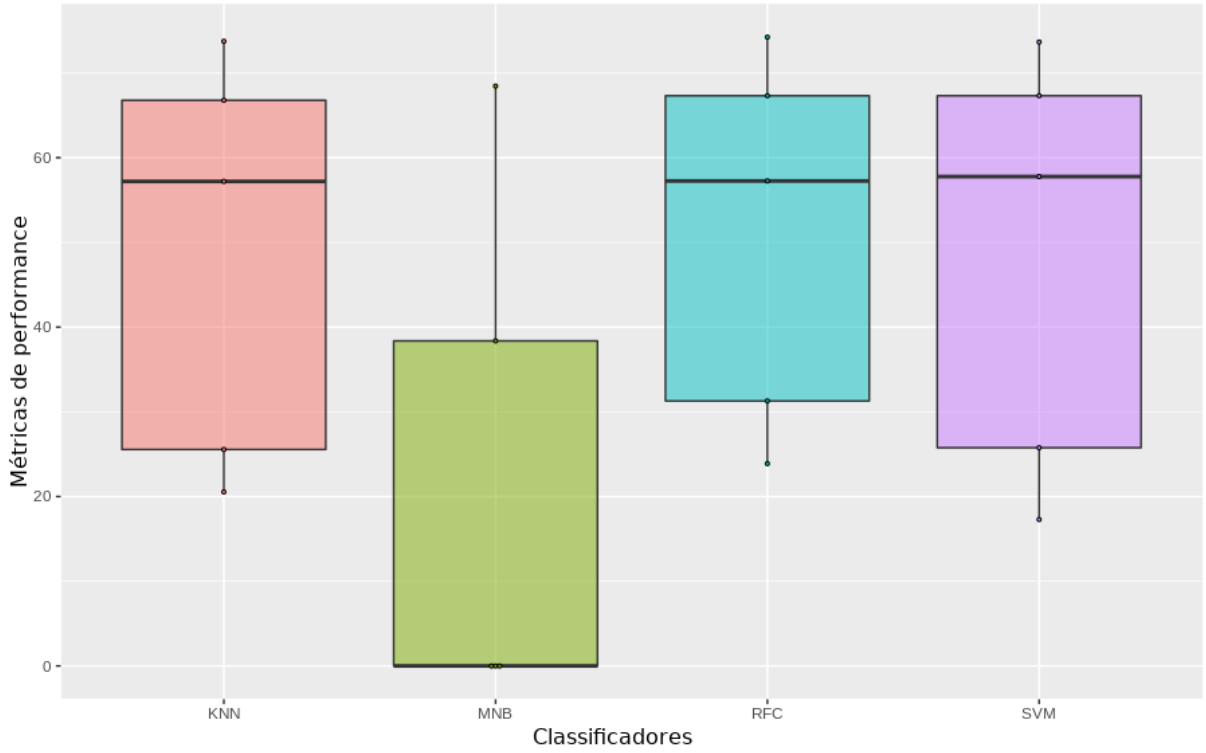

Figura 4.4: Comparação das médias de desempenho da métrica *f1 score* de cada modelo.

Não existem trabalhos publicados na área de Aprendizado de Máquina que utilizam o banco de dados do programa Precoce MS para tentar classificar o grau de gordura da carcaça bovina. Apesar da baixa taxa de acerto nas classes 1 e 5, a ocorrência delas na vida real representam $0,48\%$ e $0,02\%$ respectivamente e não apresentam, na maioria dos casos, ameça para a qualidade dos classificadores quando utilizados em dados do dia-a-dia dos produtores rurais.

Outro fator que contribui para os resultados encontrados é o fato da percepção humana ser facilmente enganada [80]. A classificação do acabamento da carcaça é feita pela medição grau da gordura da mesma. Essa medida é calculada visualmente por pessoas, chamadas de classificadores, dentro do frigorífico. A base da avaliação do grau de gordura é muitas vezes subjetiva, com atributos como aparência, textura e coloração. Consequentemente, as

classificações feitas pelo programa Precoce MS podem ter algum nível de comprometimento na precisão dos resultados.

O objetivo principal deste trabalho foi a criação de um modelo de classificação do grau de gordura da carcaça de bovinos levando em consideração os dados do programa Precoce MS. Conforme indica a Figura 4.4, esse objetivo foi alcançado mostrando que não existem diferenças significativas nos resultados dos algoritmos RFC ou SVM. Sendo assim, o modelo final pode ser gerado a partir de qualquer um desses modelos, sem risco de prejuízos no desempenho da classificação, contanto que usem os parâmetros da coluna Melhores parâmetros da Tabela 4.3.

4.1 Aplicação do modelo

Com o objetivo de aplicar o modelo, pretende-se criar um portal online pelo qual o produtor rural informará seu processo produtivo atual, ou almejado, bem como os dados do bovino que pretende levar ao abate em um frigorífico. Os dados serão, então, enviados para um serviço Web que devolverá uma resposta ao portal. A resposta devolvido mostrará para o produtor qual o grau de gordura da carcaça esperado e qual a porcentagem de acerto do modelo para a predição. Toda essa comunicação ocorrerá em formato JSON [1]. O esquema dessa modelo pode ser visto na Figura 4.5

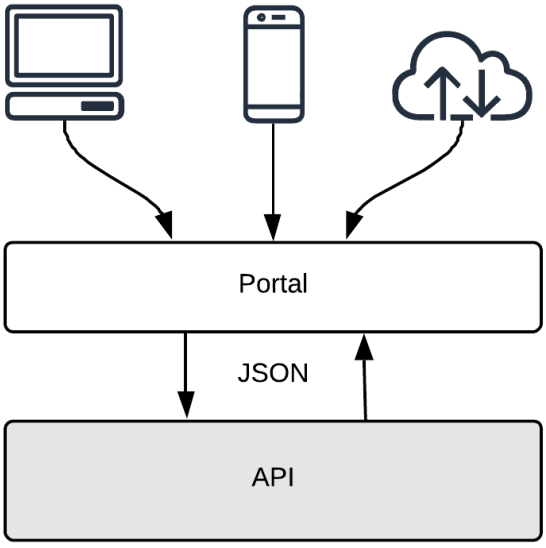

Figura 4.5: Esquema proposto da aplicação do modelo baseado em serviços Web.

Os serviços Web ou *Application Programming Interface* (API) são vistos como

[1]JavaScript Object Notation (JSON) é um formato leve para armazenar e transportar dados pela Internet, caracterizado por ser autodescritivo e fácil de entender [81].

aplicativos de negócios modulares autocontidos que possuem interfaces abertas, baseadas na Internet e baseadas em padrões [82]. Um serviço aberto significa, essencialmente, que possui uma interface publicada que pode ser chamada pela Internet.

Dessa forma, os produtores rurais, sempre que desejarem, poderão consultar o portal a fim de ter indicadores do grau de gordura da carcaça que seu gado produzirá ao ser abatido. Logo, este trabalho visa contribuir com a diminuição das incertezas relacionadas à adequação do processo produtivo e, consequentemente, auxiliá-los nas tomadas de decisões.

CAPÍTULO

5

Conclusões

5.1 Resumo dos Objetivos e Principais Resultados

Este trabalho teve o objetivo de construir um classificador do grau de gordura da carcaça para auxiliar os produtores rurais na tomada de decisão para obtenção de uma carne de melhor qualidade. Para isso, levou-se em consideração os dados do programa Precoce MS que contêm 1,05 milhão abates de bovinos e seus respectivos processos produtivos.

Os algoritmos de aprendizado de máquina testados, usando o resultado da validação cruzada, foram: Multinomial Naive Bayes (MNB), Random Forest Classifier (RFC), K-Nearest Neighbours (KNN) e Support Vector Machines (SVM). Após a otimização dos parâmetros, na qual a técnica *grid search* com validação cruzada foi utilizada, pôde-se notar uma melhora na *accuracy* de todos os modelos.

Observou-se, também, uma melhora nas métricas *precision*, *recall* e *f1 score* para algumas classes. Sendo assim, os resultados foram considerados satisfatórios, pois o modelo de classificação passou a apresentar uma taxa de confusão menor das classes 1 e 5 com outras classes. Visto que, apenas as carcaças que apresentam acabamento 2, 3 e 4 são remuneradas.

Após a execução da análise de variância ANOVA na métrica *f1 score* e o teste *post-hoc* de Tukey, dois algoritmos apresentaram uma diferença significativa no desempenho em relação aos outros: RFC e SVM. A validação dos modelos foi feita utilizando validação cruzada de 5 *folds* em um conjunto de dados de treinamento balanceado com a técnica ENN. Os resultados para a *accuracy* foram 70,45%$(+/-)0,139$ e 70,11%$(+/-)0,003$ para os algoritmos RFC e SVM,

respectivamente.

Outro fato interessante que pode ser levantado é: Quais são as características que mais influenciam no acabamento da carcaça? A resposta para essa pergunta foi obtida utilizando a técnica Recursive Feature Elimination (RFE) em conjunto com o algoritmo RFC. Portanto, as cinco características mais relevantes são: o peso da carcaça, o sexo, o mês de abate, a maturidade do animal e a localização da fazenda.

O código fonte deste trabalho pode ser encontrado no GitHub acessando o link: https://github.com/higornucci/classificacao-aulas/tree/master/dissertacao.

5.2 Dificuldades encontradas

A primeira dificuldade encontrada refere-se ao desbalanceamento do conjunto de dados de abate. As classes 1 e 5 representam $0,48\%$ e $0,02\%$ dos dados respectivamente. Muitos algoritmos de Aprendizado de Máquina não generalizam muito bem com dados desbalanceados. Consequentemente, foi necessário um estudo e comparação sobre técnicas de balanceamento de dados.

A segunda dificuldade encontrada foi a exigência de computadores potentes para processar mais de 1,05 milhão de dados. A utilização de um *cluster* de alta performance foi necessária pois o processamento dos algoritmos exigia muita memória RAM e vários processos rodando em paralelo. Portanto, foram necessários entre um e dois para saber o resultado de cada experimento.

5.3 Trabalhos Futuros

O produtor rural é bonificado apenas pelas carcaças classificadas como 2, 3 e 4. Uma otimização que pode ser feita para auxiliar na obtenção de resultados que favoreçam o produtor rural a descobrir se ele vai ou não ser bonificado e, após essa resposta, tentar mostrar qual tipo de carcaça vai ser gerado é a utilização de Aprendizado de Máquina Hierárquico [83]. Um exemplo que pode ser o ponto de partida para essa análise é separar o conjunto de dados em um classificador binário com uma classe que representa os graus de gordura de carcaças que são bonificadas e uma que representas as que não são bonificadas. Assim que descobrir se vai ou não ser bonificado, dividir novamente o conjunto de dados nas classes que representam o estado atual, ou seja, caso o classificador responda que a existe uma chance maior da carcaça ser bonificada, remover os dados que representam uma carcaça não bonificada e dividir novamente o conjunto de dados em 2, 3 e 4.

Os algoritmos utilizados nesse trabalho são chamados de “algoritmos rasos”. Existe uma área no Aprendizado de Máquina chamada de “aprendizado profundo” ou, do inglês, *Deep Learning* que permite que modelos compostos de múltiplas camadas aprendam a representar os dados com múltiplos níveis de abstração [65]. Dessa forma, acredita-se que utilizar esses algoritmos pode melhorar as métricas utilizadas e entregar um resultado mais coeso para o produtor rural.

Dois fatores importantes que podem influenciar na qualidade geral da carne produzida como a raça e a alimentação individual [13] [14] não são levados em consideração no programa Precoce MS. Além disso, os dados do processo produtivo de cada propriedade rural são atualizados anualmente. Sendo assim, o processo produtivo pode ter mudado desde o último abate até o atual e os dados podem confundir o modelo de classificação.

Por fim, este trabalho sugere que o programa Precoce MS passe a monitorar o bovino individualmente do início de sua criação até o momento da tipificação do acabamento da carcaça e passe a coletar os dados da raça do animal abatido. Consequentemente, os próximos modelos de classificação terão uma acurácia mais alta por incluir outros atributos importantes e decisivos para a variação no acabamento de carcaça.

Bibliografia

[1] IBGE. Estatística da produção pecuária, 2018. https://biblioteca.ibge.gov.br/visualizacao/periodicos/2380/epp_2018_3tri.pdf, acessado em 04/02/2019. Citado nas páginas 1 e 5.

[2] MAPA. Exportação, 2017. http://www.agricultura.gov.br/assuntos/sanidade-animal-e-vegetal/saude-animal/exportacao, acessado em 20/10/2017. Citado na página 1.

[3] Embrapa Gado de Corte. Carcaça. http://old.cnpgc.embrapa.br/publicacoes/naoseriadas/cortes/textos/carcaca.html, acessado em 23/10/2017. Citado na página 2.

[4] Pecuária e Abastecimento Ministério da Agricultura. Instrução normativa - 9, de 04/05/2004, 2004. http://www.defesaagropecuaria.sp.gov.br/www/legislacoes/popup.php?action=view&idleg=643, acessado em 23/10/2017. Citado na página 2.

[5] Pedro Eduardo de Felício. *Bovinocultura de Corte - Volumes I e II*, chapter Classificação e tipificação de carcaças bovinas, pages 1257–1276. FEALQ, 2010. Citado nas páginas 2 e 3.

[6] Pecuária e Abastecimento Ministério da Agricultura. Portaria nº 268, de 4 de maio de 1995. http://www.cidasc.sc.gov.br/inspecao/files/2012/08/PORTARIA-268.pdf, Acessado em 23/10/2017. Citado na página 2.

[7] MINISTÉRIO DA AGRICULTURA. Portaria nº 612, de 5 de outubro de 1989. http://www.cidasc.sc.gov.br/inspecao/files/2012/08/PORTARIA-MAPA-612-DE-05-10-1989.pdf, acessado em 23/10/2017. Citado na página 2.

[8] TE Lawrence, JD Whatley, TH Montgomery, and LJ Perino. A comparison of the usda ossification-based maturity system to a system based on

dentition. *Journal of animal science*, 79(7):1683–1690, 2001. Citado na página 3.

[9] Carmen Dalla Rosa Bittencourt. Classificação automática do acabamento de gordura em imagens digitais de carcaças bovinas. 2009. Citado na página 4.

[10] Resolução conjunta sefaz/sepaf nº 69 de 30 de agosto de 2016. https://www.legisweb.com.br/legislacao/?id=328328, 2016. accessed in 2017/10/23. Citado nas páginas 3, 4, e 28.

[11] Governo do Estado de Mato Grosso do Sul. Decreto nº 14.526, de 28 de julho de 2016. http://www.spdo.ms.gov.br/diariodoe/Index/Download/42493, acessado em 26/09/2017. Citado na página 4.

[12] Albino Luchiari Filho. Produção de carne bovina no brasil qualidade, quantidade ou ambas? In *Simpósio sobre Desafios e Novas Tecnologias na Bovinocultura de Corte - SIMBOI*, page 29, Brasília, DF, 2006. Citado na página 5.

[13] Carlos Guilherme A. Mielitz Netto Luciana Pötter, José Fernando Piva Lobato. Produtividade de um modelo de producao para novilhas de corte primiparas aos dois, tres, quatro anos de idade. *Revista Brasileira de Zootecnia*, 29(3):861–870, 2000. Citado nas páginas 5 e 49.

[14] Ivan Luiz Brondani, Alexandre Amstalden Moraes Sampaio, João Restle, Joilmaro Rodrigo Pereira Rosa, Cássio Vieira Marques Dos Santos, Maurício Dos Santos Fernandes, Fábio Cervo Garagorry, and Ivan Heck. Desempenho de bovinos jovens das raças aberdeen angus e hereford, confinados e alimentados com dois níveis de energia. *Revista Brasileira de Zootecnia*, 33(6 SUPPL. 3):2308–2317, 2004. Citado nas páginas 5 e 49.

[15] Eduardo Simões Corrêa, Fernando Paim Costa, Geraldo Augusto de Melo Filho, and Mariana de Aragão Pereira. Sistemas de produção melhorados para gado de corte em mato grosso do sul. Technical Report 102, Embrapa Gado de Corte, Campo Grande, MS, 2006. Citado na página 5.

[16] Alexandre Rodrigo Mendes Fernandes, Alexandre Amstalden Moraes Sampaio, Wignez Henrique, Dilermando Perecin, Emanuel Almeida De Oliveira, and Rymer Ramiz Túllio. Avaliação econômica e desempenho de machos e fêmeas Canchim em confinamento alimentados com dietas à base de silagem de milho e concentrado ou cana-de-açúcar e concentrado

contendo grãos de girassol. *Revista Brasileira de Zootecnia*, 36(4):855–864, 2007. Citado na página 5.

[17] Paulo Santana Pacheco, João Restle, José Henrique Souza da Silva, Ivan Luiz Brondani, Leonir Luiz Pascoal, Miguelangelo Ziegler Arboitte, and Aline Kellermann de Freitas. Desempenho de novilhos jovens e superjovens de diferentes grupos genéticos terminados em confinamento. *Revista Brasileira de Zootecnia*, 34:963–975, 2005. Citado na página 5.

[18] Paulo Santana Pacheco, João Restle, Fabiano Nunes Vaz, Aline Kellermann de Freitas, João Teodoro Padua, Mikael Neumann, and Miguelangelo Ziegler Arboitte. Avaliação econômica da terminação em confinamento de novilhos jovens e superjovens de diferentes grupos genéticos. *Revista Brasileira de Zootecnia*, 35(1):309–320, 2006. Citado na página 5.

[19] Mariana de Aragão Pereira. Demandas tecnológicas dos sistemas de produção de bovinos de corte no brasil – gestão da empresa rural. *Documentos / Embrapa Gado de Corte*, 219:7–20, 2016. Citado na página 5.

[20] Solange Oliveira Rezende, Jaqueline Brigladori Pugliesi, Edson Augusto Melanda, and Marcos Ferreira de Paula. *Sistemas inteligentes: fundamentos e aplicações*, chapter Mineração de dados, pages 307–335. Manole, 2005. Citado na página 9.

[21] Jiawei Han. *Data mining: Concepts and Techniques*. Morgan Kaufmann, 225 Wyman Street, Waltham, MA 02451, USA, 2012. Citado nas páginas 10, 12, e 13.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. Citado nas páginas 10, 15, e 21.

[23] UCI. Iris data set. https://archive.ics.uci.edu/ml/datasets/Car+Evaluation, acessado em 23/11/2017. Citado nas páginas 12, 21, e 23.

[24] Sebastian Raschka. *Python Machine Learning*. Packt Publishing Ltd, Liverly Place, 35, Livery Street, Birmingham B3 2PB, UK, 2015. Citado nas páginas 11, 17, 18, 20, 23, 32, e 36.

[25] Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959. Citado na página 14.

[26] Tom M. Mitchell. *Machine Learning*. WCB McGraw-Hill, 1997. Citado nas páginas 14 e 40.

[27] Stuart Russell, Peter Norvig, and Ernest Davis. *Inteligência Artificial: uma abordagem moderna*. Elsevier, Rio de Janeiro, RJ, 2013. Citado nas páginas 14 e 15.

[28] Xiaojin Zhu. *Semi-Supervised Learning*, pages 892–897. Springer US, Boston, MA, 2010. Citado na página 14.

[29] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. Citado nas páginas 16, 18, e 22.

[30] UCI. Car evaluation data set. https://archive.ics.uci.edu/ml/datasets/iris, acessado em 23/11/2017. Citado na página 17.

[31] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995. Citado na página 18.

[32] João Gama. Árvores de decisão. *Palestra ministrada no Núcleo da Ciência de Computação da Universidade do Porto, Porto*, 2002. Citado na página 19.

[33] Victor Francisco Rodriguez-Galiano, Bardan Ghimire, John Rogan, Mario Chica-Olmo, and Juan Pedro Rigol-Sanchez. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67:93–104, 2012. Citado nas páginas 19 e 37.

[34] Sebastian Raschka. Naive bayes and text classification i-introduction and theory. *arXiv preprint arXiv:1410.5329*, 2014. Citado na página 20.

[35] Harry Zhang. The optimality of naive bayes. *AA*, 1(2):3, 2004. Citado na página 20.

[36] Yudong Zhang, Siyuan Lu, Xingxing Zhou, Ming Yang, Lenan Wu, Bin Liu, Preetha Phillips, and Shuihua Wang. Comparison of machine learning methods for stationary wavelet entropy-based multiple sclerosis detection: decision tree, k-nearest neighbors, and support vector machine. *Simulation*, 92(9):861–871, 2016. Citado na página 21.

[37] Daniel Commenges and Helene Jacqmin. The intraclass correlation coefficient: distribution-free definition and test. *Biometrics*, pages 517–526, 1994. Citado na página 21.

[38] Scikit-Learn Developers. Plot the decision boundaries of a votingclassifier. http://scikit-learn.org/stable/auto_examples/ensemble/

plot_voting_decision_regions.html, acessado em 25/11/2017. Citado nas páginas 21 e 23.

[39] Enrique Vidal Ruiz. An algorithm for finding nearest neighbours in (approximately) constant average time. *Pattern Recognition Letters*, 4(3):145–157, 1986. Citado nas páginas 22 e 37.

[40] Ana C. Lorena and A.C.P.L.F. de Carvalho. Uma Introdução às Support Vector Machines. *Revista de Informática Teórica e Aplicada*, 14(2):43–67, 2007. Citado na página 23.

[41] Python Software Foundation. What is python? executive summary, 2017. https://www.python.org/doc/essays/blurb/, accessed in 2017/11/16. Citado na página 24.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. Citado na página 24.

[43] Institut National de Recherche en Informatique et en Automatique. Site do inria. https://www.inria.fr/, acessado em 16/11/2017. Citado na página 24.

[44] scikit-learn developers. About us. http://scikit-learn.org/stable/about.html, Acessado em 16/11/2017. Citado na página 24.

[45] Fernando Maia da Mota. Uma abordagem de análises olap e de mineração de dados para suporte à tomada de decisão no setor da pecuária de corte do brasil. Master's thesis, Universidade Federal de Mato Grosso do Sul, 2016. Citado na página 24.

[46] Jason D. M. Rennie and Ryan Rifkin. Improving Multiclass Text Classification with the Support Vector Machine. *Massachusetts Institute of Technology AI Memo 2001-026*, (October 2001):1–14, 2001. Citado na página 25.

[47] Aruna Govada, Bhavul Gauri, and S K Sahay. Distributed Multi Class SVM for Large Data Sets. In *Proceedings of the Third International Symposium on Women in Computing and Informatics*, WCI '15, pages 54–58, New York, NY, USA, 2015. ACM. Citado na página 25.

[48] Ürün Do\vgan, Tobias Glasmachers, and Christian Igel. A Unified View on Multi-class Support Vector Classification. *J. Mach. Learn. Res.*, 17(1):1550–1831, 2016. Citado na página 25.

[49] J Weston and C Watkins. Support Vector Machines for Multi-Class Pattern Recognition. *Proceedings of the 7th European Symposium on Artificial Neural Networks (ESANN-99)*, (Abril):219–224, 1999. Citado na página 25.

[50] Yudong Zhang, Siyuan Lu, Xingxing Zhou, Ming Yang, Lenan Wu, Bin Liu, Preetha Phillips, and Shuihua Wang. Character classification framework based on support vector machine and k-nearest neighbour schemes. *ScienceAsia*, 42(1):46–51, 2016. Citado na página 25.

[51] Kusworo Adi, Sri Pujiyanto, Oky Dwi Nurhayati, and Adi Pamungkas. Beef quality identification using color analysis and k-nearest neighbor classification. In *2015 4th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, pages 180–184. IEEE, nov 2015. Citado na página 25.

[52] K K Chaturvedi and V B Singh. An empirical comparison of machine learning techniques in predicting the bug severity of open and closed source projects. *International Journal of Open Source Software and Processes*, 4(2):32–59, 2012. Citado na página 25.

[53] Meera Sharma, Punam Bedi, K. K. Chaturvedi, and V. B. Singh. Predicting the priority of a reported bug using machine learning techniques and cross project validation. *International Conference on Intelligent Systems Design and Applications, ISDA*, pages 539–545, 2012. Citado na página 26.

[54] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016. Citado na página 30.

[55] Aurlien Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 1st edition, 2017. Citado nas páginas 30 e 33.

[56] Perfil estatístico de mato grosso do sul. Technical report, Secretaria de Estado de Meio Ambiente e Desenvolvimento Econômico (Semade), 2016. http://www.seinfra.ms.gov.br/wp-content/uploads/sites/6/2017/06/Perfil_Estatístico_MS_2016.pdf, accessed in 2018/12/12. Citado na página 30.

[57] Estações do ano no brasil. https://www.calendarr.com/brasil/estacoes-do-ano/. accessed in 2018/12/12. Citado na página 31.

[58] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003. Citado nas páginas 31 e 42.

[59] Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12):2270–2285, 2005. Citado na página 32.

[60] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 875–886. Springer, 2009. Citado na página 33.

[61] Ricardo Barandela, Rosa M Valdovinos, J Salvador Sánchez, and Francesc J Ferri. The imbalanced training sample problem: Under or over sampling? In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 806–814. Springer, 2004. Citado na página 33.

[62] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421, 1972. Citado na página 33.

[63] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. Citado na página 33.

[64] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004. Citado na página 33.

[65] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. Citado nas páginas 34 e 49.

[66] James T Townsend. Theoretical analysis of an alphabetic confusion matrix. *Perception & Psychophysics*, 9(1):40–50, 1971. Citado na página 34.

[67] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European Conference on Information Retrieval*, pages 345–359. Springer, 2005. Citado na página 36.

[68] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution.

In *Pattern recognition (ICPR), 2010 20th international conference on*, pages 3121–3124. IEEE, 2010. Citado nas páginas 36 e 39.

[69] Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence*, pages 488–499. Springer, 2004. Citado na página 37.

[70] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001. Citado na página 37.

[71] Maha Asiri, Hamid Nemati, and Fereidoon Sadri. Feature reduction improves classification accuracy in healthcare. In *Proceedings of the 22Nd International Database Engineering & Applications Symposium*, IDEAS 2018, pages 193–198, New York, NY, USA, 2018. ACM. Citado na página 40.

[72] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2017. Citado na página 40.

[73] Asha Gowda Karegowda, MA Jayaram, and AS Manjunath. Feature subset selection problem using wrapper approach in supervised learning. *International journal of Computer applications*, 1(7):13–17, 2010. Citado na página 41.

[74] Marc Johannes, Jan C Brase, Holger Fröhlich, Stephan Gade, Mathias Gehrmann, Maria Fälth, Holger Sültmann, and Tim Beißbarth. Integration of pathway knowledge into a reweighted recursive feature elimination approach for risk stratification of cancer patients. *Bioinformatics*, 26(17):2136–2144, 2010. Citado na página 41.

[75] Pablo M Granitto, Cesare Furlanello, Franco Biasioli, and Flavia Gasperi. Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products. *Chemometrics and Intelligent Laboratory Systems*, 83(2):83–90, 2006. Citado na página 41.

[76] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014. Citado na página 41.

[77] Vladimir N Vapnic. Statistical learning theory. *A Wiley-Interscience Publication*, 1998. Citado na página 42.

[78] Øyvind Langsrud. Anova for unbalanced data: Use type ii instead of type iii sums of squares. *Statistics and Computing*, 13(2):163–167, 2003. Citado na página 43.

[79] Daniel MacRae Keenan. A tukey nonadditivity-type test for time series nonlinearity. *Biometrika*, 72(1):39–44, 1985. Citado na página 44.

[80] F. J. Francis. Colour quality evaluation of horticultural crops. *HortScience*, 15(1):14–15, 1980. Citado na página 44.

[81] Nicolas Serrano, Josune Hernantes, and Gorka Gallardo. Service-oriented architecture and legacy systems. *IEEE software*, 31(5):15–19, 2014. Citado na página 45.

[82] Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. Web services. In *Web Services*, pages 123–149. Springer, 2004. Citado na página 46.

[83] Martin Pelikan and David E Goldberg. A hierarchy machine: Learning to optimize from nature and humans. *Complexity*, 8(5):36–45, 2003. Citado na página 48.