UNIVERSIDADE FEDERAL DE MINAS GERAIS

Habib Asseiss Neto

METODOLOGIA DE APRENDIZADO AUTOML BASEADO EM INFORMAÇÕES DE COMPLEXIDADE DE INSTÂNCIAS

Belo Horizonte
2020

HABIB ASSEISS NETO

METODOLOGIA DE APRENDIZADO AUTOML BASEADO EM INFORMAÇÕES DE COMPLEXIDADE DE INSTÂNCIAS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Sérgio Vale Aguiar Campos
Coorientador: Ronnie Cley de Oliveira Alves

Belo Horizonte
Dezembro de 2020

Asseiss Neto, Habib.

A844m Metodologia de aprendizado AutoML baseado em informações de complexidade de instâncias [manuscrito] / Habib Asseiss Neto. — 2020.
xvii, 95 f.; il.; 29cm.

Orientador: Sérgio Vale Aguiar Campos.
Coorientador: Ronnie Cley de Oliveira Alves.

Tese (doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 89-95

1. Computação – Teses. 2. Aprendizado do computador. – Teses. 3. Redes neurais convolucionais. – Teses. 4 Teoria de Resposta ao Item – Teses. 5. Leite - Análise -Teses. I. Campos, Sérgio Vale Aguiar. II. Alves, Ronnie Cley de Oliveira. III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. IV. Título.

CDU 519.6*82(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz CRB - 6ª Região nº 819.

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Metodologia de aprendizado AutoML baseado em informações de complexidade de instâncias

HABIB ASSEISS NETO

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. SÉRGIO VALE AGUIAR CAMPOS - Orientador
Departamento de Ciência da Computação - UFMG

PROF. RONNIE CLEY DE OLIVEIRA ALVES - Coorientador
DS - Instituto Tecnológico Vale

PROF. ADRIANO CÉSAR MACHADO PEREIRA
Departamento de Ciência da Computação - UFMG

PROF. LEORGES MORAES DA FONSECA
Departamento de Tecnologia e Inspeção em Produtos de Origem Animal - UFMG

PROF. NALVO FRANCO DE ALMEIDA JUNIOR
Faculdade de Computação - UFMS

PROF. WALDEYR MENDES CORDEIRO DA SILVA
Departamento de Informática - IFECTG

Belo Horizonte, 7 de Dezembro de 2020.

Resumo

A análise de dados biológicos é uma tarefa importante, pois permite a obtenção de informações úteis e da expansão do conhecimento sobre determinado domínio biológico. Materiais biológicos podem ser analisados por diversas técnicas e um método amplamente utilizado para analisar a composição estrutural desses materiais é denominada análise de espectroscopia por infravermelho, que permite extrair informações através da emissão da luz infravermelha nas amostras.

As técnicas de espectroscopia produzem um grande volume de dados, o que tornam complexas análises manuais por especialistas. A Ciência da Computação, através do Aprendizado de Máquina, pode ajudar nessa tarefa, oferecendo maneiras de compreender e produzir conhecimentos importantes a partir de amostras espectrais. As Redes Neurais Convolucionais, geralmente aplicadas com sucesso no reconhecimento de imagens, são especificamente adequadas para os dados espectrais das amostras, uma vez que as estruturas dos espectros podem ser vistas como uma imagem.

Este trabalho conduz experimentos nos quais se procura detectar a ocorrência de adulteração no leite bovino através de análises de espectroscopia por infravermelho, utilizando, para isso, uma arquitetura proposta de rede neural convolucional e modelos *ensemble* de árvores de decisão. Nos experimentos realizados, dados espectrais de milhares de amostras, puras e adulteradas, do leite bovino foram submetidas à rede neural convolucional proposta, aos modelos *ensemble* e a outros métodos comumente utilizados para este fim, permitindo uma comparação de diferentes abordagens. A abordagem proposta foi capaz de detectar adulterantes com acurácia de até 98,76% para os métodos de rede neural convolucional e *ensemble* de árvores, enquanto os métodos de *baseline* comumente utilizados produziram acurácias médias de 65,88%.

Apesar do bom desempenho da rede neural convolucional para o problema abordado, elaborar arquiteturas de redes neurais que ofereçam bom desempenho para problemas genéricos é uma tarefa desafiadora. Geralmente, a busca por uma arquitetura adequada é um processo específico para o problema abordado e é conduzido por cientistas especializados através de testes manuais e extensivos, além de ser necessário

conhecimento prévio em problemas semelhantes abordados anteriormente. O Aprendizado de Máquina Automatizado, ou AutoML, pode colaborar nesse processo, pois tem como um de seus objetivos buscar as arquiteturas mais adequadas para o problema fornecido como entrada de forma completamente automatizada, sem intervenção humana.

No entanto, métodos de Aprendizado de Máquina, ou mesmo de AutoML, geralmente não levam em consideração características individuais das instâncias que fazem parte dos conjuntos de dados analisados. Ao considerar informações que refletem características de cada amostra, pode-se avaliar a complexidade e a habilidade dos métodos. A Teoria de Resposta ao Item (IRT) é uma abordagem da área de psicometria que pode ser adaptada ao Aprendizado de Máquina, podendo oferecer descrições de complexidade no nível das instâncias, além de caracterizar habilidades inerentes aos modelos.

Neste trabalho, propõe-se uma metodologia inovadora baseada em AutoML e IRT, denominada NASirt, capaz de oferecer uma maior explicabilidade de modelos de Redes Neurais Convolucionais. O método seleciona automaticamente um conjunto de modelos com diferentes arquiteturas e submete, a cada modelo, instâncias específicas do conjunto de dados, baseando-se em informações fornecidas pela IRT. O NASirt pode determinar as arquiteturas de redes neurais convolucionais mais adequadas para um determinado problema e realizar a classificação de instâncias com acurácias médias maiores que outros métodos analisados. Além disso, por utilizar os conceitos de IRT, o NASirt pode determinar a complexidade de instâncias do conjunto de dados analisado e estimar os níveis de habilidade de modelos com diferentes arquiteturas, obtendo um avanço na explicabilidade de modelos.

Diversos experimentos foram conduzidos para avaliar o comportamento e a viabilidade da metodologia em conjuntos de dados reais, comparando os resultados com outros métodos de *benchmark*. Os resultados mostram que o método proposto apresenta desempenho, na maioria dos casos, melhor que os métodos comparados. A metodologia proposta foi capaz de gerar acurácias médias de 96,96% para um conjunto de dados, enquanto um modelo de rede neural convolucional criado manualmente apresentou 78,43%, uma abordagem de votação com centenas de modelos apresentou 69,06% e um sistema AutoML já existente apresentou 91,81% de acurácia.

Palavras-chave: Aprendizado de Máquina, AutoML, Teoria de Resposta ao Item, Redes Neurais Convolucionais.

Abstract

Analysis of biological data is a very important task since it allows one to obtain useful information and to expand knowledge about some biological domain. Biological materials can be analyzed by several techniques and a widely used method is the infrared spectroscopy analysis, which allows the information extraction through the emission of infrared light in the samples.

Spectroscopy techniques produce a large volume of data, which make manual analysis by experts complex. Computer Science, through Machine Learning, can help in this task, offering ways to understand and produce important knowledge from spectral samples. Convolutional Neural Networks, generally successfully applied to image recognition, are specifically suitable for spectral data of samples, since spectral structure can be seen as an image.

This work conducts experiments in order to detect the occurrence of adulteration in bovine milk through infrared spectroscopy analyzes using a proposed convolutional neural network architecture and ensemble decision trees models. In the experiments, spectral data from thousands of pure and adulterated samples of bovine milk were subjected to the proposed convolutional neural network, to the ensemblee models and to other commonly used methods for this purpose, allowing a comparison of different approaches. The proposed approach was able to detect adulterants with an accuracy of up to 98.76% for convolutional neural network and tree embedding, while the commonly used baseline methods produced average accuracy of 65.88%.

Despite the good performance of the convolutional neural network for the milk problem, designing neural network architectures that offer good performance for generic problems is a challenging task. Generally, the search for an adequate architecture is a specific process for the problem addressed and it is conducted by specialized scientists through manual and extensive tests, and also prior knowledge on similar problems is required. Automated Machine Learning, or AutoML, can collaborate in this process, since its objectives is the search for the most adequate architectures for the input problem in an automated way and no human interaction.

However, Machine Learning methods, or even AutoML methods, generally do not consider individual characteristics from instances that belong to the analyzed datasets. By considering information that reflects each sample's characteristics, it is possible to evaluate the methods' habilities. Item Response Theory (IRT) is a psychometrics approach that can be adapted to Machine Learning and can offer complexity descriptions on an instance level, and also characterize inherit abilities to Machine Learning models.

In this work, we propose an innovative methodology based on AutoML and IRT that is capable of offering high explicability of Convolutional Neural Network models. The method selects the most adequate instances to be submitted to different models, based on the difficulty and discrimination information and also models abilities. Several experiments were conducted in order to evaluate the methodology viability on real datasets, comparing the results with other benchmark methods. Results show that the proposed method presents, in most cases, better performance over the other methods. The proposed methodology was capable of generating average accuracies of 96.96% for a specific dataset, while a Convolutional Neural Network manually created presented 78.43%, a voting approach with hundreds of models presented 69.06% and an already existing AutoML system presented 91.81% of accuracy.

Keywords: Machine Learning, AutoML, Item Response Theory, Convolutional Neural Networks.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

A análise da composição de materiais biológicos pode oferecer conhecimentos valiosos para diversas áreas da ciência. O estudo de dados biológicos é, muitas vezes, complexo e a Ciência da Computação oferece metodologias e ferramentas que tornam viável a sua análise. Uma das técnicas de extração de informações de materiais é a espectroscopia no infravermelho, baseada em operações simples e não destrutivas, que emite raios de luz e obtém coordenadas espectrais que representam composições químicas presentes no material.

A espectroscopia no infravermelho é uma técnica que mede a absorção da luz infravermelha por um material e é amplamente utilizada para análise de materiais biológicos, com diversas características interessantes, entre elas o fato de ser não destrutiva do ponto de vista do material e o baixo custo do processo [Baker et al., 2014]. Em geral, componentes de sistemas biológicos complexos, tais como tecido animal ou vegetal, podem ser caracterizados por essa técnica [Stuart, 2012]. Com os dados obtidos, diversas abordagens para análise podem ser empregadas, como a análise de dados multivariada ou a utilização de métodos de aprendizado de máquina, como redes neurais artificiais [Santos et al., 2013; Baker et al., 2014; Liu et al., 2017].

O leite bovino é um exemplo de material comumente submetido ao processo de espectroscopia no infravermelho. Por se tratar de um alimento de grande importância para o homem e de amplo consumo comercial, o leite bovino tem diversos parâmetros de qualidade analisados antes de sua comercialização. A espectroscopia pode analisar diferentes características de qualidade do leite, como os teores de gorduras, lactose e outras proteínas [Gondim et al., 2017]. Em geral, o espectro obtido do leite pode ser utilizado para a detecção de adulterantes presentes no material, mas as técnicas mais utilizadas são as análises estatísticas multivariadas ou métodos de regressão [Botelho et al., 2015; Gondim et al., 2017].

Por se tratarem de coordenadas numéricas que representam a interação da luz com o material biológico, os espectros obtidos pela espectroscopia podem ser explorados computacionalmente. Para transformar os dados obtidos em informações úteis, diferentes técnicas computacionais podem ser empregadas, mais especificamente, métodos de aprendizado de máquina, que têm como característica a detecção de padrões e a descoberta de conhecimento em conjuntos de dados não estruturados. Esses métodos têm ganhado ampla atenção em diversos temas nas últimas décadas e os avanços obtidos são significativos. Redes neurais artificiais, especialmente as redes neurais convolucionais, vêm ganhando espaço crescente na área de aprendizado de máquina, com propostas de arquiteturas específicas para diferentes habilidades, como reconhecimento de imagens, processamento de linguagem natural, predições de dados temporais como de mercados financeiros, entre outros [Witten et al., 2016].

Uma das abordagens recentes e que têm apresentado resultados interessantes no aprendizado de máquina é a combinação de diferentes modelos para a obtenção de uma capacidade preditiva mais robusta [Polikar, 2006]. Esta abordagem é denominada meta-aprendizado (*meta-learning*) e se caracteriza como um modelo cujas predições são baseadas nos resultados outros modelos de aprendizado de máquina. O meta-aprendizado está inserido numa área denominada Aprendizado de Máquina Automatizado, conhecida simplesmente por AutoML, que é uma uma abordagem moderna para a obtenção automatizada de arquiteturas de modelos, procedimentos de treinamentos e otimização de hiperparâmetros para problemas específicos. Este tema vem ganhado ampla atenção devido à sua capacidade de gerar modelos adequados para diferentes domínios, sem a necessidade de testes e ajustes manuais e extensivos.

Recentemente, foi demonstrado que analisar individualmente a complexidade das amostras dos conjuntos de dados pode aumentar o desempenho de métodos de aprendizado de máquina [Smith et al., 2014]. A Teoria de Resposta ao Item (*Item Response Theory* - IRT) é uma abordagem clássica da área de psicometria que avalia questionários e indivíduos e pode determinar as características de complexidade dos itens de um questionário, bem como as habilidades dos indivíduos analisados. A IRT pode ser trazida para o aprendizado de máquina através de uma associação de indivíduos para modelos de aprendizado de máquina, e de itens de um questionário para instâncias de um conjunto de dados. O uso de IRT permite estimar as habilidades dos modelos de aprendizado de máquina considerando as instâncias aos quais eles são submetidos [Martínez-Plumed et al., 2019].

Como parte deste trabalho, diferentes métodos de aprendizado de máquina foram analisados em experimentos realizados em parceria com o Laboratório de Análise da Qualidade do Leite da Escola de Veterinária da Universidade Federal de Minas Gerais,

onde foram obtidas milhares de amostras do leite bovino, que foram submetidas a diferentes análises. Inicialmente, considerando o problema da adulteração do leite, foram realizadas adulterações controladas em um subconjunto das amostras com substâncias comumente utilizadas em práticas de adulteração. As amostras foram, então, submetidas ao processo de espectroscopia no infravermelho e os dados produzidos foram coletados para análises computacionais. Dentre os dados obtidos, estão presentes as coordenadas dos espectros infravermelhos de cada amostra, que foram utilizadas como entrada para diferentes métodos de aprendizado de máquina.

A análise desses dados permitiu que fosse proposta uma arquitetura de rede neural especialmente projetada para o reconhecimento desses espectros. Além disso, métodos *ensemble* de árvores de decisão conhecidos foram utilizados, como comparação, com dados de composição das amostras produzidas pelo equipamento de espectroscopia. Os resultados dos experimentos mostram que arquiteturas de rede convolucional apresentam desempenho superior em todos os casos, quando comparados aos métodos *ensemble* de árvores, e também a métodos estatísticos tradicionalmente empregados nesse tipo de análise.

Como a arquitetura de rede neural proposta para o reconhecimento de adulteração no leite é específica para o problema abordado, não é possível prever o comportamento desse modelo específico para conjuntos de dados diferentes. O AutoML pode ajudar nesta questão, pois oferece formas de generalizar a criação de um modelo para dados de outros problemas e domínios. Além disso, a utilização de IRT pode caracterizar mais adequadamente as instâncias analisadas pelo método e, assim, oferecer uma abordagem inovadora na área de AutoML.

Portanto, propõe-se, neste trabalho, o desenvolvimento de uma metodologia baseada em AutoML que realiza uma busca por arquiteturas de redes neurais convolucionais a fim de determinar automaticamente modelos específicos capazes de gerar os melhores resultados para o conjunto de dados analisado. A metodologia proposta, denominada NASirt, baseia-se na obtenção de características de complexidade de instâncias e de habilidade de modelos fornecidas pela IRT. O método submete automaticamente instâncias específicas do conjunto de dados a um grupo de modelos de redes neurais artificiais com diferentes arquiteturas, baseando-se em informações fornecidas pela IRT. O NASirt pode determinar as arquiteturas de redes neurais convolucionais mais adequadas para um determinado problema e realizar a classificação de instâncias com acurácias médias maiores que outros métodos analisados. Além disso, por utilizar os conceitos de IRT, o NASirt pode determinar a complexidade de instâncias do conjunto de dados analisado e estimar os níveis de habilidade de modelos com diferentes arquiteturas, obtendo um avanço na explicabilidade de modelos, especificamente modelos

de redes neurais artificiais, onde sua compreensão exata ainda é desafiadora.

A metodologia proposta foi submetida a diferentes experimentos com conjuntos de dados distintos. Além do conjunto de dados de adulterantes do leite, amostras distintas acrescidas manualmente com soro do leite foram analisadas, bem como amostras provenientes de tecidos de diferentes espécies de árvores. Nos experimentos, o método NASirt foi avaliado com a utilização de dois parâmetros, dificuldade e discriminação, além de uma abordagem de votação com modelos definidos pelo método. As execuções do NASirt foram comparadas com diferentes métodos de *benchmark*: o modelo de rede neural convolucional descrito no problema da adulteração do leite, uma abordagem de votação com centenas de modelos gerados e, por fim, um sistema AutoML existente, denominado Auto-Keras.

Os resultados indicam que o método proposto apresenta desempenho superior aos métodos comparados na maioria dos casos individuais executados. Considerando os valores médios de acurácia para as diferentes execuções em cada experimento, o método NASirt apresentou acurácias superiores a todos os métodos de *benchmark*. Para o conjunto de dados com as espécies de árvores, por exemplo, a metodologia proposta foi capaz de gerar acurácias médias de 96,96%, enquanto o modelo de rede neural convolucional criado manualmente apresentou 78,43%, a abordagem de votação com centenas de modelos apresentou 69,06% e o sistema Auto-Keras apresentou 91,81% de acurácia.

1.1 Motivação

Modelos de aprendizado de máquina são, muitas vezes, considerados "caixa-preta". Isto é, entender a relação entre as funções realizadas por cada componente de uma rede neural convolucional com a decisão tomada na predição final do modelo é um desafio. A combinação de modelos realizada por métodos de meta-aprendizado torna ainda mais complexa a análise de suas decisões.

Diversos são os problemas em que a estrutura de dados a ser analisada é baseada em coordenadas espectrais. Esses dados espectrais, tais como os dados obtidos por análises biológicas, geralmente são repletos de informação a ser explorada, e são adequados às redes neurais convolucionais devido ao seu formato semelhante a imagens. Essa característica, portanto, motivou a utilização desse tipo de rede neural na metodologia proposta.

Os esforços para a obtenção de um modelo de aprendizado de máquina especializado para a avaliação da adulteração no leite motivaram a elaboração de uma me-

todologia de AutoML. A metodologia proposta pode permitir a obtenção de modelos de redes neurais convolucionais adequados para outros conjuntos de dados espectrais. A fim de oferecer melhores capacidades preditivas, a abordagem utiliza informações de complexidade de instâncias e de habilidades de modelos baseadas na Teoria de Resposta ao Item. Além disso, a abordagem baseada em IRT permite o conhecimento dos limites dos modelos utilizados e dos limiares de suas capacidades de classificação.

1.2 Objetivos

1.2.1 Objetivo geral

O objetivo geral deste trabalho é desenvolver uma metodologia de Aprendizado de Máquina Automatizado baseada nos conceitos da Teoria de Resposta ao Item para problemas de classificação cujos conjuntos de dados sejam obtidos a partir de análises espectrais. A metodologia proposta, denominada NASirt, pode determinar tanto a complexidade das instâncias dos conjuntos de dados quanto as habilidades de modelos de aprendizado de máquina, gerando automaticamente arquiteturas de redes neurais convolucionais capazes de realizar predições com altos níveis de acurácia.

1.2.2 Objetivos específicos

- Extrair e analisar dados espectrais de amostras do leite bovino, permitindo-se a execução de modelos de aprendizado de máquina para a caracterização das amostras e a inferência da ocorrência de adulterações no leite.
- Propor uma metodologia que utiliza diferentes modelos de aprendizado de máquina para estudo e análise específicos de amostras espectrais de leite bovino, a fim de se detectar a ocorrência de adulteração por substâncias diversas no material através da realização de espectroscopia no infravermelho.
- Propor uma nova arquitetura de rede neural convolucional capaz de extrair informações das coordenadas espectrais obtidas sem necessidade de preprocessamentos, como ocorre com outras técnicas.
- Propor uma metodologia baseado em AutoML que busca por melhores arquiteturas de redes neurais convolucionais, que utiliza a informação de dificuldades das instâncias de IRT e que seja capaz descrever a capacidade de aprendizado dos modelos, o que pode permitir um avanço significativo na área da interpretabilidade de modelos de aprendizado de máquina.

- Avaliar e comparar a metodologia proposta com outras abordagens semelhantes, apresentando as capacidades preditivas de cada método através de diversos experimentos.

1.3 Contribuições do trabalho

Nesta seção são descritas as principais contribuições deste trabalho.

- O trabalho aborda o problema de classificação através da proposta de uma metodologia para a detecção de adulterantes no leite bovino a partir de dados obtidos por análises de espectroscopia no infravermelho. A adulteração no leite é detectada com alta precisão tanto para a determinação específica de uma entre cinco substâncias testadas, quanto simplesmente para determinar a ocorrência ou não da adulteração. A técnica apresentada aborda métodos baseados em árvores de decisão e redes neurais convolucionais.

- Propõe-se uma arquitetura de rede neural convolucional especializada, capaz de extrair informações de coordenadas espectrais sem necessidade de preprocessamentos. A rede neural proposta realiza a extração automática de características diretamente do espectro, utilizando os mesmos conceitos no reconhecimento de imagens, tarefa na qual as redes convolucionais são empregadas mais comumente. A arquitetura proposta apresenta alto desempenho e a capacidade de reconhecer características das curvas espectrais de forma mais precisa que outros métodos.

- Como contribuição principal deste trabalho, apresenta-se o desenvolvimento de uma metodologia de AutoML inovadora, denominada NASirt, que realiza uma busca por arquiteturas automaticamente e pode determinar a complexidade de instâncias e habilidade de modelos. O NASirt utiliza os conceitos da Teoria de Resposta ao Item para oferecer um nível de explicabilidade para automatização da modelagem de arquiteturas de redes neurais, especificamente para problemas de classificação e para conjuntos de dados oriundos de análises espectrais.

1.3.1 Contribuições adicionais

- Artigo publicado em periódico internacional: Asseiss Neto, H.; Tavares, W. L.; Ribeiro, D. C.; Alves, R. C. O.; Fonseca, L. M. & Campos, S. V. A. **On the utilization of deep and ensemble learning to detect milk adulteration**.

BioData Mining, 13, 2019. ISSN 1756-0381. DOI: 10.1186/s13040-019-0200-5. Disponível em `https://biodatamining.biomedcentral.com/articles/10.1186/s13040-019-0200-5`.

- Artigo submetido para o periódico internacional *Expert Systems with Applications*, atualmente disponível no arXiv: Asseiss Neto, H.; Alves, R. C. O. & Campos, S. V. A. (2020) **NASirt: AutoML based learning with instance-level complexity information**. *arXiv e-prints*: 2008.11846. Disponível em `https://arxiv.org/abs/2008.11846`.

- Repositório público contendo o código desenvolvido para a metodologia proposta, bem como exemplos de conjuntos de dados utilizados como entrada e arquivos de saída gerados pelo código. Disponível em `https://osf.io/mrqc3/`.

1.4 Organização do texto

A seguir, descreve-se, brevemente, a estrutura do texto para o restante dos capítulos do trabalho.

No Capítulo 2, apresentam-se conceitos e definições gerais que envolvem técnicas de análise de espectroscopia no infravermelho e análise da qualidade do leite, bem como oferece uma visão geral de métodos de aprendizado de máquina, detalhando árvores de decisão, métodos *ensemble* e redes neurais. Por fim, abordam-se técnicas de avaliação de modelos, separações de conjuntos de dados para treinamento e teste, validação cruzada e métricas de desempenho para problemas de classificação e regressão.

O Capítulo 3 apresenta um estudo que avalia a ocorrência de adulteração no leite bovino através da aplicação de métodos de aprendizado de máquina em dados obtidos pela espectroscopia no infravermelho. Propõe-se uma arquitetura de rede neural convolucional, normalmente utilizada para reconhecimento de imagens, para a detecção e extração de características de amostras espectrais sem necessidade de preprocessamentos. Por fim, diferentes métodos são comparados e os resultados são apresentados.

No Capítulo 4, inicialmente descrevem-se conceitos mais detalhados de AutoML e da Teoria de Resposta ao Item. Posteriormente, é proposta a metodologia de AutoML, que envolve a combinação de modelos de aprendizado de máquina, utilizando conceitos de IRT, como a dificuldade de instâncias e habilidades dos modelos. Os detalhes da implementação da metodologia são apresentados, bem como a execução de testes para determinar o desempenho do método e a comparação com métodos de *benchmark*. São apresentadas, ainda, análises de complexidade da metodologia proposta.

Por fim, no Capítulo 5, apresenta-se a conclusão e as propostas de continuidade do trabalho.

Capítulo 2

Referencial Teórico

Neste capítulo serão apresentados conceitos e técnicas do Aprendizado de Máquina, uma subárea da Inteligência Artificial amplamente abordada em estudos da Ciência da Computação. Apresentam-se detalhes de métodos bastante utilizados, como árvores de decisão e redes neurais artificiais e suas variações, além de análises estatísticas consideradas "tradicionais" como Análise de componentes principais (PCA) e Regressão por mínimos quadrados parciais (PLS). Aborda-se, também, técnicas para avaliação de modelos de aprendizado de máquina e formas de medição de desempenho desses modelos.

2.1 Aprendizado de máquina

Aprendizado de máquina (ou *machine learning*) é o estudo computacional onde são desenvolvidos algoritmos baseados em métodos estatísticos que, a partir de conjuntos de dados de entrada, aprendem e reconhecem padrões desses dados. Algoritmos de aprendizado de máquina têm ganhado ampla atenção nas últimas décadas devido a sua capacidade de contribuir na descoberta de conhecimento de praticamente quaisquer temas. Considerada uma subárea da Inteligência Artificial, o aprendizado de máquina tem sido aplicado em diversas áreas da ciência e na indústria como um todo, apresentando resultados extremamente interessantes em aplicações como reconhecimento de imagens e de voz, serviços financeiros, análise de sentimentos, serviços médicos, entre outros [Alpaydin, 2014; Kubat, 2017].

No aprendizado de máquina, modelos computacionais são construídos com base em teorias estatísticas que analisam dados e têm como objetivo principal realizar inferências e tomar decisões a partir desses dados. Os problemas de aprendizado de

máquina podem ser categorizados em três paradigmas principais: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.

No aprendizado supervisionado, o objetivo é o aprendizado de um mapeamento entre os dados de entrada e saída com base em valores corretos fornecidos por um supervisor (dados rotulados). Os dados rotulados são submetidos a um algoritmo que aprende regras e mapeia as instâncias, avaliando seu desempenho com base nos rótulos fornecidos. Em outras palavras, o aprendizado supervisionado envolve a construção de um modelo estatístico com objetivo de predizer a resposta para observações futuras (predição) ou melhor entender o relacionamento entre a resposta e as entradas (inferência) [James et al., 2017]. São exemplos de métodos de aprendizado supervisionado *Support Vector Machines* (SVM), *K-Nearest Neighbor* (KNN), Árvores de Decisão, Redes Neurais Artificiais, entre outros.

No aprendizado não supervisionado, não há a presença de um supervisor que rotula os dados com valores corretos. Os algoritmos recebem apenas os dados de entrada e o objetivo é encontrar regularidades e simetrias nos dados a partir de padrões que possam ser detectados nesses dados [Alpaydin, 2014]. Em estatística, este aprendizado pode ser denominado estimativa de densidade, e um método bastante conhecido é o agrupamento (*clustering*), em que se busca encontrar grupos de instâncias semelhantes, dado um conjunto de dados de entrada.

Por fim, o aprendizado por reforço compreende métodos em que um agente aprende como atingir um objetivo complexo através da execução de certos passos de forma a maximizar uma recompensa numérica cumulativa. O algoritmo do aprendizado por reforço é treinado para descobrir quais ações tomadas levam à maior recompensa. As ações tomadas por um algoritmo não afetam apenas a recompensa mais imediata, como também todas as recompensas subsequentes das próximas situações. Essas duas características, busca por tentativa e erro e recompensa em atraso, são as mais importantes do aprendizado por reforço [Sutton & Barto, 2018].

2.2 Aprendizado supervisionado

O aprendizado supervisionado é atualmente o paradigma estudado na maioria das pesquisas atuais da área de aprendizado de máquina [Sutton & Barto, 2018]. Este paradigma envolve métodos que aprendem a partir de um conjunto de dados de treinamento cujos registros são rotulados previamente. No processo denominado treinamento, os métodos tentam descobrir o relacionamento entre um ou mais atributos de entrada (chamados de variáveis independentes) e um atributo alvo (chamado de

variável dependente). O relacionamento descoberto é representado em uma estrutura denominada *modelo*, que é uma função inferida a partir do treinamento e geralmente descreve e explica fenômenos que estão ocultos no conjunto de dados. Os modelos podem posteriormente ser usados para a predição do valor do atributo alvo a partir do valor dos atributos de entrada [Maimon & Rokach, 2010].

O aprendizado supervisionado pode ser dividido em modelos de regressão e modelos de classificação (classificadores). Na regressão, os modelos mapeiam o espaço de entrada em um valor contínuo, caracterizando um registro do conjunto de dados de forma quantitativa. A idade ou a altura de um indivíduo, o valor de uma casa ou o valor de uma ação no mercado financeiro são exemplos de variáveis quantitativas que podem ser induzidas a partir de um modelo de regressão. Em outras palavras, a saída esperada é uma variável numérica (variável dependente) como uma função da entrada (variáveis independentes). Na regressão a variável de saída é sempre uma variável contínua, de valor real, e, na prática, refere-se a quantidades, medidas e tamanhos a serem preditos pelos problemas [James et al., 2017].

Por outro lado, os classificadores mapeiam o espaço de entrada em classes predefinidas, identificando a qual categoria um novo registro pertence. A classificação determina um registro de forma qualitativa. Exemplos de variáveis qualitativas incluem o gênero de uma pessoa (masculino ou feminino), a marca de um produto (marca X ou Y), se um cliente possui dívidas (sim ou não) ou o tipo de câncer diagnosticado em um paciente [James et al., 2017]. Em outras palavras, a saída esperada é uma variável categórica (variável dependente) como uma função da entrada (variáveis independentes).

Um modelo de aprendizado supervisionado é inferido a partir do processo denominado treinamento, onde é inferido, ou encaixado (*fit*), com base nos dados de treinamento. As capacidades preditivas do modelo são avaliadas em um processo posterior, denominado teste, utilizando um conjunto de dados separado do treinamento. Os detalhes de treinamento e teste serão abordados na Seção 2.4, a seguir.

Diferentes métodos podem induzir modelos de predição baseados no aprendizado supervisionado. Em alguns casos, os métodos podem ser adequados para ambos os problemas, classificação ou regressão. De forma geral, é possível transformar os problemas de regressão em classificação, adequando a extensão da variável alvo contínua em conjuntos de intervalos que serão usados como classes discretas [Torgo & Gama, 1996]. A seguir serão abordados alguns métodos de aprendizado de máquina amplamente utilizados.

2.2.1 Árvores de decisão

Árvores de decisão constroem um modelo de classificação ou regressão expresso como uma partição recursiva no espaço de instâncias. A estrutura criada é uma árvore composta de raiz, de nós internos e de folhas, também chamadas de nós terminais ou nós de decisão. Cada nó interno da árvore divide o espaço de instâncias em dois ou mais espaços de acordo com uma função discreta dos atributos de entrada. Cada folha representa uma decisão no atributo alvo (valor numérico ou uma classe). Dado um modelo de árvore de decisão, uma predição é realizada para as instâncias navegando-se pela árvore da raiz a uma folha, de acordo com o resultado dos testes pelo caminho dos nós internos.

O algoritmo base para a construção de uma árvore de decisão é denominado ID3 e seu sucessor é denominado C4.5 [Maimon & Rokach, 2010]. Um outro algoritmo bastante utilizado é o CART [Hastie et al., 2017]. Um modelo de árvore é induzido com base nas instâncias do conjunto de dados de entrada, processo denominado treinamento. A construção da árvore é feita de forma recursiva. De forma geral, inicialmente seleciona-se um atributo para se tornar o nó raiz e cria-se uma ramificação da raiz para cada possível valor do atributo. Isso separa o conjunto de instâncias em subconjuntos, um para cada valor do atributo. Depois, o processo pode ser repetido recursivamente para cada ramificação, usando apenas as instâncias correspondentes àquela ramificação. Quando todas as instâncias de um nó têm a mesma predição, o desenvolvimento dessa parte da árvore para [Witten et al., 2016].

A construção de uma árvore de decisão leva em conta os conceitos de entropia e ganho de informação. A ideia geral do cálculo de entropia é, se as instâncias em um subconjunto têm todas um mesmo valor (subconjunto homogêneo), a entropia é 0 e se as instâncias têm seus valores igualmente distribuídos, a entropia é 1. O ganho de informação é baseado na diminuição da entropia após uma subdivisão do conjunto em um atributo (ramificação do nó). O atributo escolhido que irá gerar uma ramificação na árvore é aquele que retornar o maior ganho de informação, isto é, o que gerar a ramificação com subconjuntos mais homogêneos. No caso de problemas de regressão, o desvio padrão é usado para calcular a homogeneidade de uma amostra numérica. Se uma amostra numérica é completamente homogênea, seu desvio padrão é 0. Na árvore com regressão, o atributo escolhido é o que apresentar a maior redução no desvio padrão [Witten et al., 2016].

2.2.2 Redes neurais artificiais

Modelos de aprendizado de máquina denominados Redes Neurais Artificiais (*Artificial Neural Networks*, ANN) são inspirados no funcionamento do cérebro. Os objetivos desses modelos, no entanto, não é entender o funcionamento do cérebro em si, mas o de construir modelos que possam se assemelhar à capacidade humana de reconhecimento de padrões. Como parte do aprendizado supervisionado, as ANN "aprendem" regras considerando os exemplos fornecidos como parte do treinamento. Uma rede neural é formada por nós conectados, denominados neurônios artificiais, ou unidades, e cada conexão transmite o sinal de um neurônio ao outro (semelhante à sinapse biológica). O neurônio possui pesos que são utilizados para ponderar as entradas recebidas e produzir uma saída. Um neurônio recebe um sinal, processa-o e então transmite uma saída para outros neurônios conectados a ele. Em geral, o sinal em um neurônio é representado por um número real e o processamento é a computação de uma função não linear da combinação de suas entradas [Chollet et al., 2015; Alpaydin, 2014].

A estrutura de neurônio, juntamente com suas entradas, pesos e saída é denominada *perceptron*. Um *perceptron*, em si, é um classificador binário linear, que decide o valor de uma saída binária de acordo com uma combinação linear baseada nos pesos das entradas. Em uma rede neural artificial, os *perceptrons* se organizam em camadas, e elas se caracterizam, geralmente, em uma camada de entrada, uma camada de saída e uma ou mais camadas internas (também chamadas de camadas ocultas). Este tipo de arquitetura de rede neural é denominado *Multilayer Perceptron* (MLP) [Alpaydin, 2014].

Em uma rede neural, as conexões entre *perceptrons* de diferentes camadas são ponderadas pelos pesos associados, que representam a força de conexão entre as unidades. As conexões são os resultados das funções de ativação dos *perceptrons* de uma camada, que são usados como entrada para outra camada. As funções de ativação podem ser a função sigmoide, ReLU (*Rectified Linear Unit*), *Softmax*, entre outras. Quando utilizadas funções não lineares, a rede neural é capaz de resolver problemas não lineares [Chollet et al., 2015].

As redes neurais devem ser treinadas com conjuntos de dados rotulados a fim de realizar predições. O treinamento envolve a execução das entradas do conjunto de treinamento na rede e, para cada entrada, sua saída é observada. Cada saída é comparada com o rótulo original e os valores de erro são utilizados em uma função de perda (*loss*) utilizada para atualizar os pesos dos neurônios da rede, em uma fase de retropropagação. Este processo é conhecido como *backpropagation* [Witten et al., 2016].

Algumas arquiteturas de redes neurais artificiais vêm ganhando popularidade de acordo com habilidades específicas, como Redes Neurais Convolucionais, para o reconhecimento de imagens, Redes Neurais Recorrentes e *Long Short-Term Memory*, para processamento de séries temporais ou linguagem natural. Em geral, redes neurais utilizam várias camadas internas, independente de sua arquitetura, e, por isso, são conhecidas pelo termo "aprendizagem profunda" (*deep learning*).

Redes neurais convolucionais

As Redes Neurais Convolucionais, ou *Convolutional Neural Networks* (CNN ou ConvNet) são uma arquitetura de rede neural muito popular e extremamente bem-sucedida em áreas como o reconhecimento de padrões em imagens e vídeos, reconhecimento de fala, classificação de textos, processamento de linguagem natural, entre outras [Bhandare et al., 2016]. As CNN são inspiradas no processamento visual do cérebro humano e possuem camadas adicionais em relação às redes neurais tradicionais.

A arquitetura geral da CNN consiste em uma ou mais camadas convolucionais, camadas de ativação não linear, camadas de agrupamento (*pooling*) e camadas de normalização. As camadas convolucionais consistem em um conjunto de filtros que aprendem, durante o treinamento, através da passagem dos dados de entrada em cada filtro. Cada filtro aprende a detectar alguma característica importante dos dados de treinamento. As camadas convolucionais são posteriormente ativadas por alguma função não linear, como a *Rectified Linear Unit* (ReLU) [Maas et al., 2013]. As CNN também podem incluir camadas de *pooling* entre camadas convolucionais a fim de reduzir a representação espacial dos dados e a complexidade da rede. Além disso, camadas de normalização podem ser utilizadas para ajustar a escala dos dados. Por fim, a última camada de convolução passa a saída para camadas de *Multilayer Perceptrons* e o funcionamento até a última camada se dá como nas redes neurais tradicionais [Witten et al., 2016].

As CNN são comumente aplicadas a dados de imagens, interpretados como *arrays* bidimensionais. Nesse caso, as camadas convolucionais detectam bordas e outras características que representam regiões importantes da imagem. A mesma ideia pode ser aplicada para dados unidimensionais, como séries temporais e dados espectrais [Liu et al., 2017].

Redes neurais recorrentes

Redes Neurais Recorrentes, ou *Recurrent Neural Networks* (RNN), são um tipo de redes neurais que se aproveitam de informações sequenciais e são particularmente apropriadas

para análises de sequências, como séries temporais, textos e áudios. Ao contrário das redes neurais tradicionais, em que um nó é considerado independente de outros nós, os nós de uma RNN possuem conexões para si mesmo ou conexões para nós em camadas anteriores. As redes recorrentes executam repetidamente a mesma tarefa para cada elemento de uma sequência, e as saídas são dependentes dos valores obtidos em computações anteriores. As conexões adicionais da RNN fazem com que elas possuam uma "memória de curto prazo" (*short-term memory*), que captura informações sobre o que já foi calculado anteriormente [Alpaydin, 2014].

Uma variação de Redes Neurais Recorrentes muito popular é denominada *Long Short-Term Memory* (LSTM), que são mais apropriadas para captura de dependências temporais relativas à sequência analisada. A principal diferença da LSTM está na estrutura denominada *célula*, considerada a memória da rede, que toma como entrada o estado anterior e a entrada atual. Internamente, a célula decide quais informações armazenar ou remover da memória. A célula, então, combina o estado anterior e a memória atual com a entrada. Esse processo é muito eficaz na captura de dependências de longo prazo nos dados de entrada [Witten et al., 2016].

2.2.3 Métodos ensemble

Métodos de aprendizado de máquina *ensemble*, também chamados de sistemas de preditores múltiplos, consistem em um conjunto de modelos treinados individualmente, cujas decisões são, de alguma forma, combinadas [Marqués et al., 2012]. O raciocínio por trás dos métodos *ensemble* é que modelos de aprendizado de máquina em geral com bom desempenho nos dados de treinamento não necessariamente oferecem um bom desempenho de generalização quando o modelo é aplicado a dados desconhecidos. Além disso, um conjunto de classificadores com desempenhos de treinamento semelhantes podem ter diferentes desempenhos de generalização "na vida real". Assim, combinar a saída de vários preditores pode reduzir o risco de escolha de um modelo com baixo desempenho para um caso específico e, portanto, técnicas de *ensemble* são bem aceitas e oferecem bons desempenhos gerais. De fato, foi demonstrado que em muitos casos os *ensemble* produzem predições mais precisas que os modelos individuais que os compõem [Marqués et al., 2012; Dong et al., 2020]. Como os métodos *ensemble* contam com a combinação de modelos, eles constroem fronteiras de decisão mais suaves, capazes de encontrar uma combinação ótima de *features* e de modelos para realizar a predição [Polikar, 2006].

Um método *ensemble* bastante popular é denominado *Bagging*, ou *Booststrap Aggregating*. O método obtém subconjuntos de dados por amostragem para o treina-

mento dos modelos a partir do conjunto de dados original. Então, as predições dos modelos individuais são agregadas, selecionando o resultado ou pelo voto da maioria dos modelos ou pela média de todos os resultados, formando uma predição final para o modelo *ensemble*. O método *Bagging* é particularmente interessante quando se tem um conjunto de dados de tamanho limitado, uma vez que a amostragem realizada garante diversidade nas instâncias para treinamento. Uma variação do algoritmo de *bagging*, que utiliza especificamente árvores de decisão, é a chamada *Random Forest* (Floresta Aleatória), criada a partir de modelos de árvores de decisão, cujos parâmetros de treinamento variam aleatoriamente. No *Random Forest*, a aleatoriedade no treinamento pode ser tanto nos dados selecionados quanto na escolha de subconjuntos de atributos (*features*) [Polikar, 2012].

Outro método *ensemble*, denominado *Boosting*, também treina modelos utilizando diferentes conjuntos de treinamento, mas os modelos são treinados para aprender sequencialmente, com cada modelo tentando minimizar o erro do modelo anterior. A combinação de preditores individualmente fracos cria um modelo com melhor desempenho [Hastie et al., 2017]. Mais especificamente, os conjuntos de dados reamostrados são construídos com o objetivo de gerar aprendizados complementares e a importância do voto é ponderado com base no desempenho de cada modelo, em vez da atribuição de mesmo peso para todos os votos [Marqués et al., 2012]. Diferentes algoritmos implementam o método de *Boosting*, como LightGBM e XGBoost, e, de forma geral, são denominados *Gradient Boosting Machines* (GBM) ou *Gradient Boosting Decision Tree* (GBDT) [Chen & Guestrin, 2016; Ke et al., 2017].

Por fim, os métodos *ensemble* Votação (*Voting*) e Generalização Empilhada (*Stacked Generalization*, ou simplesmente *Stacking*), são mais amplos e podem combinar modelos de diferentes algoritmos de aprendizado de máquina (modelos heterogêneos), ao contrário dos métodos *Bagging* e *Boosting* que combinam modelos de um mesmo tipo. No *Voting*, os modelos são treinados independentemente e seus resultados são combinados através de alguma forma de votação, como a votação da maioria (*majority voting*) simples ou ponderada. No *Stacking*, modelos são treinados individualmente e combinados através do treinamento de um meta-modelo, que produz a predição final tomando como entrada as saídas dos submodelos [Polikar, 2012]. Por exemplo, para um problema de classificação, pode-se definir como submodelos um classificador KNN, uma regressão logística e uma árvore de decisão, e, para a combinação, cria-se uma rede neural como meta-modelo. Nessa estrutura, a rede neural toma como entrada as saída dos três classificadores "fracos", será treinada e aprenderá a retornar as saídas baseadas nesses resultados.

2.3 Análises estatísticas tradicionais

Embora os conceitos de aprendizado de máquina utilizem fortemente estatística, pode-se observar perspectivas distintas em ambas as áreas. Na estatística tradicional, a transição de observações particulares para descrições genéricas é chamada de inferência, enquanto o aprendizado é chamado de estimativa. Durante muito tempo, os métodos estatísticos eram quase que exclusivamente modelos lineares, pois a inferência de modelos não lineares era inviável antes de os computadores tornarem-se baratos e abundantes [Alpaydin, 2014]. Além disso, esses métodos eram aplicados apenas a pequenas amostras de dados. Conforme a computação evolui, o estudo de métodos estatísticos mais complexos se torna mais comum. Na atualidade, é impraticável analisar os dados manualmente devido à enorme quantidade de dados disponíveis. Assim, existe um interesse crescente em modelos computacionais capazes de analisar e extrair informações dos dados. Atualmente, estatística, aprendizado de máquina, inteligência artificial, mineração de dados, processamento de sinais e outras áreas se misturam em suas definições e abordagens [Alpaydin, 2014].

2.3.1 Regressão linear

Embora alguns métodos estatísticos clássicos possam ser muito básicos quando comparados aos métodos de aprendizado de máquina mais modernos, eles ainda são úteis e amplamente utilizados para o aprendizado estatístico [James et al., 2017]. A regressão linear é uma abordagem simples para o aprendizado supervisionado, em particular, interessante para a predição de uma resposta quantitativa. A aplicação deste método é interessante pois pode ser um ponto de partida para outras abordagens. De fato, muitas outras abordagens estatísticas são extensões ou generalizações da regressão linear [James et al., 2017].

A regressão linear simples é a predição do relacionamento linear entre duas variáveis contínuas, uma variável independente e uma variável dependente. No entanto, em muitas aplicações, existe mais de um fator que potencialmente influencia a saída. Nesses casos, a regressão linear múltipla pode descrever como um atributo alvo depende linearmente de um conjunto de atributos de entrada [Witten et al., 2016].

A regressão linear, assim como qualquer outro método de regressão, pode ser utilizada para problemas de classificação através da execução da regressão para cada classe, definindo a saída igual a 1 para instâncias que pertencem à classe, e 0 para aquelas que não pertencem. O resultado é uma expressão linear para a classe. Então, dado um conjunto de teste com classes desconhecidas, deve-se calcular o valor de cada

expressão linear e escolher aquela que obteve o maior valor. Essa operação pode ser entendida como uma aproximação de uma função numérica de "pertencimento" para cada classe: a função retorna 1 para instâncias que pertençam à classe, e 0 para outras instâncias [Witten et al., 2016].

2.3.2 Regressão logística

A regressão logística é uma extensão da regressão linear para problemas de classificação. A técnica de transformação da regressão linear para classificação discutida anteriormente calcula valores para classes que não podem ser interpretados como probabilidades, pois os valores são apenas 0 ou 1, e o valor de predição é uma interpolação linear entre os pontos da entrada. A solução para a classificação, neste caso, é utilizar a função logística, método conhecido como regressão logística [Witten et al., 2016].

A regressão logística modela uma variável dependente binária a partir de variáveis independentes utilizando a função logística (sigmoide). Neste modelo, a predição é uma probabilidade de o valor da entrada ser entre 0 e 1. Portanto, a regressão logística é uma alternativa viável à versão de classificação da regressão linear, e é um modelo muito popular devido à sua simplicidade e sua habilidade de inferir afirmações estatísticas sobre os dados [Kuhn & Johnson, 2013].

2.3.3 Análise de componentes principais (PCA)

A Análise de Componentes Principais (*Principal Component Analysis*, PCA) é uma técnica de preprocessamento que reduz os dados, gerando um conjunto menor de variáveis que visam capturar a maioria das informações nas variáveis originais. Esta técnica é denominada redução de dados, e com sua aplicação, poucas variáveis podem ser usadas para oferecer uma representação razoável ao conjunto de dados original. O PCA converte um conjunto de variáveis correlacionadas em um conjunto de valores linearmente não correlacionados denominados componentes principais (*Principal Components*, PC). A transformação é realizada de forma que a primeira componente principal tenha a maior variância possível, isto é, uma componente que seja responsável pela maior variabilidade nos dados possível. Outras componentes principais são calculadas de forma que a maior variabilidade restante seja capturada [Kuhn & Johnson, 2013].

O PCA é um método não supervisionado, pois o cálculo das componentes principais não leva em conta os valores de atributo alvo. O método tem como vantagem a criação de componentes não correlacionadas, úteis para a aplicação em alguns modelos

de predição. O PCA é geralmente utilizado como uma ferramenta de análise exploratória dos dados e sua operação pode ser entendida como a de revelar a estrutura interna dos dados de forma que melhor explique a variância desses dados [Kuhn & Johnson, 2013].

2.3.4 Regressão por mínimos quadrados parciais (PLS)

A regressão por mínimos quadrados parciais (*Partial Least Squares*, PLS) é semelhante à análise PCA, mas em vez de encontrar componentes com a máxima variância entre a variável alvo e as variáveis independentes, o método PLS encontra um modelo de regressão linear através da projeção das variáveis dependentes e independentes para um novo espaço. O PLS encontra componentes que maximizam a variância das variáveis independentes, enquanto simultaneamente exige que essas componentes tenham máxima correlação com o atributo alvo. Assim, como utiliza as variáveis dependentes ou alvo, o PLS é considerado um método de aprendizado supervisionado [Kuhn & Johnson, 2013].

O método de PLS possui a vantagem, em relação ao PCA, de reduzir a dimensionalidade das variáveis necessariamente produzindo novas variáveis que explicam o atributo alvo. O PCA não considera qualquer aspecto da variável dependente ao selecionar suas componentes. O PLS, por outro lado, deriva as componentes enquanto considera a variável alvo e gera uma solução de regressão linear com variáveis correlacionadas [Kuhn & Johnson, 2013].

O PLS atualmente é muito usado na Quimiometria e na Bioinformática, além de outras áreas relacionadas [Alpaydin, 2014].

2.4 Avaliação de modelos

Nesta seção serão discutidas a separação de conjuntos de dados de treinamento, validação e teste, técnicas de validação cruzada e, por fim, serão abordadas métricas de desempenho para avaliação do poder preditivo de modelos de aprendizado de máquina.

2.4.1 Separação de conjuntos de dados

Em geral, métodos de aprendizado de máquina têm seu desempenho avaliado utilizando uma separação de conjuntos de dados de forma padronizada: conjuntos de treinamento, conjunto de validação e conjuntos de teste. A razão para essa separação é que, se houvesse um treinamento com um modelo de aprendizado de máquina e seu desempenho

fosse avaliado com os mesmos dados, obter-se-ia uma estimativa incondizente com a realidade prática desse modelo, nesse caso, seria um modelo com viés (*bias*) indesejado. Deve-se sempre buscar um modelo que seja capaz de funcionar bem quando aplicado a instâncias desconhecidas [Alpaydin, 2014; Arlot & Celisse, 2010].

Os procedimentos envolvidos em um trabalho com aprendizado de máquina normalmente envolvem fases distintas que utilizam essas divisões específicas do conjunto de dados. A primeira é o treinamento do modelo, um processo que induz um modelo que se ajusta aos dados (operação comumente denominada *fit*). Durante o treinamento, o modelo pode utilizar um conjunto de validação para avaliar internamente suas capacidades e direcionar ajustes de parâmetros na indução do modelo. Em alguns casos, a etapa de validação é omitida e o conjunto de dados de validação não existe. Por fim, o modelo deve ser testado em um conjunto de dados cujas instâncias são desconhecidas do modelo. O conjunto de dados de teste é usado para estimar o desempenho final do modelo e deve ser mantido completamente separado dos outros dados [James et al., 2017; Russell & Norvig, 2009].

Quando se tem um conjunto suficientemente grande de dados, a melhor abordagem para avaliar um modelo é aleatoriamente dividir o conjunto de dados em três partes, para as fases descritas anteriormente: treinamento, validação e teste [Hastie et al., 2017]. Uma proporção típica é separar 50% dos dados para treinamento, 25% para validação e 25% para teste, mas esses valores variam de acordo com o estudo, o domínio dos dados e o autor. Deve-se decidir, de forma sensata, como dividir as proporções dos conjuntos, dada a quantidade de dados disponíveis [Kuhn & Johnson, 2013].

2.4.2 Validação cruzada

A divisão do conjunto de dados em subconjuntos específicos é importante para o treinamento e a avaliação correta do modelo. Porém, um conjunto de teste independente oferece apenas uma avaliação do modelo e apresenta restrições ao caracterizar incerteza nos resultados, o que é importante para a generalização de um modelo. Além disso, conjuntos independentes de teste, quando muito grandes, separam os dados de forma a aumentar o viés nas estimativas de desempenho do teste do modelo. As técnicas de validação cruzada (*cross-validation*) em geral diminuem o viés das estimativas de desempenho e oferecem uma visão mais realista do poder preditivo dos modelos. É comum que não se aborde um conjunto de validação específico quando se trata de validação cruzada, apenas conjuntos de treinamento e teste [Kuhn & Johnson, 2013].

Validação cruzada pode ser dividida em dois tipos principais: exaustiva e não

exaustiva. Na validação cruzada exaustiva, um modelo de aprendizado de máquina é treinado e testado de todas as formas possíveis de dividir o conjunto de dados original em conjuntos de treinamento e teste. Na validação cruzada não exaustiva, não são computadas todas as formas de separação dos conjuntos de dados e são, em geral, mais utilizadas na prática. As técnicas mais comuns de validação cruzada não exaustiva são: validação cruzada *k-fold* e método *hold-out*.

Validação cruzada k-fold

Nesta forma de validação cruzada, chamada de *k-fold cross-validation*, o conjunto de dados original é dividido aleatoriamente em k subconjuntos de mesmos tamanhos aproximados. Um modelo é inferido (treinado) usando todas as instâncias exceto as da primeira partição ($k = 1$). As instâncias separadas são usadas para teste e estimativas de desempenho do modelo. O procedimento se repete com a seleção das instâncias da segunda partição ($k = 2$) para o teste, enquanto o modelo é treinado com todas as instâncias remanescentes, e assim por diante. As estimativas dos k testes são sumarizadas utilizando-se, normalmente, a média e o desvio padrão de cada execução [Kuhn & Johnson, 2013; Hastie et al., 2017].

Uma variação comum deste método é denominada *stratified k-fold cross-validation*, que seleciona k partições de forma que o valor do atributo alvo médio é aproximadamente igual em todas as dobras (*folds*). Uma outra variação é denominada *leave-one-out cross-validation*, que é o caso especial onde k é o número total de instâncias do conjunto de dados [Maimon & Rokach, 2010].

Em geral, a escolha de k é geralmente 5 ou 10, mas não há regra formal [Kuhn & Johnson, 2013]. Neste método, todas as instâncias são usadas tanto para treinamento quanto para teste, e cada instância é usada para teste exatamente uma vez.

Validação cruzada hold-out

O método *hold-out* é o mais básico de validação cruzada, onde simplesmente se atribui aleatoriamente as instâncias do conjunto de dados a dois subconjuntos: treinamento e teste. O tamanho desses subconjuntos é arbitrário, mas, em geral, o conjunto de treinamento é maior que o conjunto de teste. Então, o modelo é treinado e testado com os subconjuntos correspondentes. Ao contrário da validação cruzada *k-fold*, em que os resultados são obtidos de uma média de execuções, o método *hold-out* envolve uma única execução isolada. O método não é indicado quando não há um número suficientemente grande de instâncias no conjunto de dados, pois pode-se obter resultados não realistas, dependendo da divisão dos subconjuntos [Kohavi, 1995].

2.4.3 Medidas de desempenho

Diferentes medidas de desempenho são consideradas para analisar a capacidade preditiva de modelos de regressão e de classificação. Para entender os pontos fortes e fracos de um modelo, pode-se avaliar diferentes métricas, visualizar os dados e assegurar que o modelo foi inferido para o propósito que foi projetado.

Modelos de regressão

Quando o atributo alvo é uma variável numérica contínua, o método mais comum para caracterizar as capacidades preditivas de um modelo é usar a raiz do erro quadrático médio (*root mean squared error*, RMSE). Esta métrica é uma função dos resíduos do modelo de regressão, que são as predições do modelo subtraídas dos valores observados. O erro quadrático médio (*mean squared error*, MSE) é calculado através do quadrado dos resíduos e suas somas. O RMSE é, então, calculado pela raiz quadrada do MSE. Esse cálculo geralmente é interpretado como o quão distante os resíduos estão de zero, na média. O RMSE pode ser entendido também como a distância média entre os valores observados e os valores preditos [Kuhn & Johnson, 2013].

Outra métrica comum é o coeficiente de determinação, geralmente apresentado como R^2. Esse valor pode ser interpretado como a proporção de informação nos dados que é explicada pelo modelo. Então, um valor de R^2 de, por exemplo, 0.75, implica que o modelo de regressão pode explicar três quartos da variação no valor esperado de saída. Para o cálculo do R^2, deve-se encontrar o coeficiente de correlação entre os valores observados e preditos (normalmente denotados por R), e depois elevá-lo ao quadrado [Kuhn & Johnson, 2013].

Modelos de classificação

Em geral, modelos de classificação produzem dois tipos de predição. Assim como modelos de regressão, os modelos de classificação produzem um valor contínuo como predição, que geralmente é usado como probabilidade, isto é, os valores preditos são o de pertencimento de instâncias a classes e estão entre 0 e 1. Além desse valor contínuo, os modelos de classificação geram uma predição de classe, considerada uma categoria discreta. Para as aplicações em geral, uma predição de categoria discreta é necessária para tomar uma decisão.

Embora os modelos de classificação produzam ambos os tipos de predição, normalmente o foco é na predição discreta de classes. No entanto, a estimativa de probabilidade fornecida para cada classe pode ser muito útil para a determinação da con-

fiança do modelo para uma determinada predição de classe [Kuhn & Johnson, 2013]. Por exemplo, em uma aplicação de detecção de *spam* nos *e-mails*, uma determinada instância (mensagem de *e-mail*) foi predita com a classe *spam* com probabilidade de 0.51, e uma outra instância foi classificada como *spam* com probabilidade de 0.99. Ambas as instâncias foram classificadas da mesma forma pelo modelo, mas o programa de *e-mail* pode usar a informação de probabilidade para avisar ao usuário das diferentes confianças na sua classificação.

Uma maneira comum de se descrever a predição de classes de um modelo de classificação é através da matriz de confusão. Esta matriz é uma tabela simples que indica os valores observados e os valores preditos pelo modelo. A Tabela 2.1 mostra um exemplo quando o valor esperado tem duas classes, a ocorrência e a não ocorrência de um determinado evento. A primeira linha da matriz corresponde a amostras preditas como ocorrência do evento, algumas são preditas corretamente (verdadeiros positivos, VP) e algumas são classificadas incorretamente (falsos positivos, FP). De forma análoga, a segunda linha da matriz contém os valores de predições de não ocorrência do evento, algumas predições corretas (verdadeiros negativos, VN) e outras predições incorretas (falsos negativos, FN). As células na diagonal principal denotam os casos onde as classes são corretamente preditas, enquanto as células fora da diagonal mostram o número de erros para cada caso possível.

Tabela 2.1. Matriz de confusão para um problema genérico de classificação de duas classes (*ocorrência* e *não ocorrência* de um evento). As células da tabela indicam o número de verdadeiros positivos (VP), falsos positivos (FP), verdadeiros negativos (VN) e falsos negativos (FN).

Predição	Observação	
	Ocorrência	Não ocorrência
Ocorrência	VP	FP
Não ocorrência	FN	VN

Na classificação, a métrica mais simples é a acurácia (precisão), um valor que representa uma taxa entre as classes observadas e preditas. Apesar de ser amplamente utilizada, a acurácia deve ser usada com cautela, pois é uma métrica que não leva em conta o tipo de erro que o modelo está fazendo. No caso do filtro de *spam*, por exemplo, o custo de remover uma mensagem marcada como *spam* incorretamente é provavelmente maior do que permitir que uma mensagem que seja *spam* passe pelo filtro. Nas situações em que os custos são diferentes, a acurácia pode não medir características importantes do modelo.

No caso de uma classificação binária (quando há duas classes possíveis), algumas métricas adicionais podem ser relevantes: sensibilidade e especificidade. A sensibilidade de um modelo é a proporção em que um evento de interesse é predito corretamente para todas as amostras em que esse evento ocorre. Em outras palavras, a sensibilidade é a taxa de verdadeiros positivos e é dada pela fórmula

$$Sensibilidade = \frac{número\ de\ amostras\ com\ o\ evento\ e\ preditas\ como\ tendo\ o\ evento}{número\ de\ amostras\ com\ o\ evento}$$

Já a especificidade é definida como a proporção das amostras em que há a não ocorrência de um evento para todas as amostras em que esse evento não ocorre. A especificidade é a taxa de verdadeiros negativos, dada pela fórmula

$$Especificidade = \frac{número\ de\ amostras\ sem\ o\ evento\ e\ preditas\ como\ não\ evento}{número\ de\ amostras\ sem\ o\ evento}$$

Existe normalmente uma relação de compromisso entre a sensibilidade e a especificidade. Intuitivamente, o aumento da sensibilidade de um modelo pode provocar perda de especificidade, já que mais amostras estão sendo preditas como ocorrência de evento. Essa troca entre sensibilidade e especificidade pode ser interessante quando há penalidades diferentes associadas com cada tipo de erro. No caso do filtro de *spam*, por exemplo, existe geralmente uma tendência para a especificidade, pois a maioria das pessoas estão dispostas a aceitar receber algumas mensagens de *spam*, desde que *e-mails* importantes não sejam bloqueados inadequadamente pelo filtro. Uma técnica que avalia a troca e a relação de compromisso é denominada curva *Receiver Operating Characteristic* (ROC) [Kuhn & Johnson, 2013].

Receiver Operating Characteristic (ROC) e Area Under the ROC Curve (AUC)

Avaliar as probabilidades de classe pode oferecer maiores informações sobre as predições do modelo do que analisar simplesmente o valor da classe. Curvas ROC utilizam a probabilidade de classificação de cada classe e consideram um intervalo de limiares de decisão sobre essas probabilidades. A curva ROC é calculada, no intervalo crescente dos limiares, através da plotagem dos valores da taxa de verdadeiros positivos (sensibilidade) e da taxa de falsos positivos ($1 -$ especificidade) [Fawcett, 2006]. A Figura 2.1 apresenta uma curva ROC de um modelo genérico, onde dois pontos diferentes são destacados, mostrando suas especificidades e sensibilidades.

A curva ROC pode ser usada para uma avaliação quantitativa de modelos. A área sob a curva ROC, representada pela sigla AUC (*Area Under the ROC Curve*), oferece uma medida condensada de desempenho relacionada a todos os possíveis limiares de

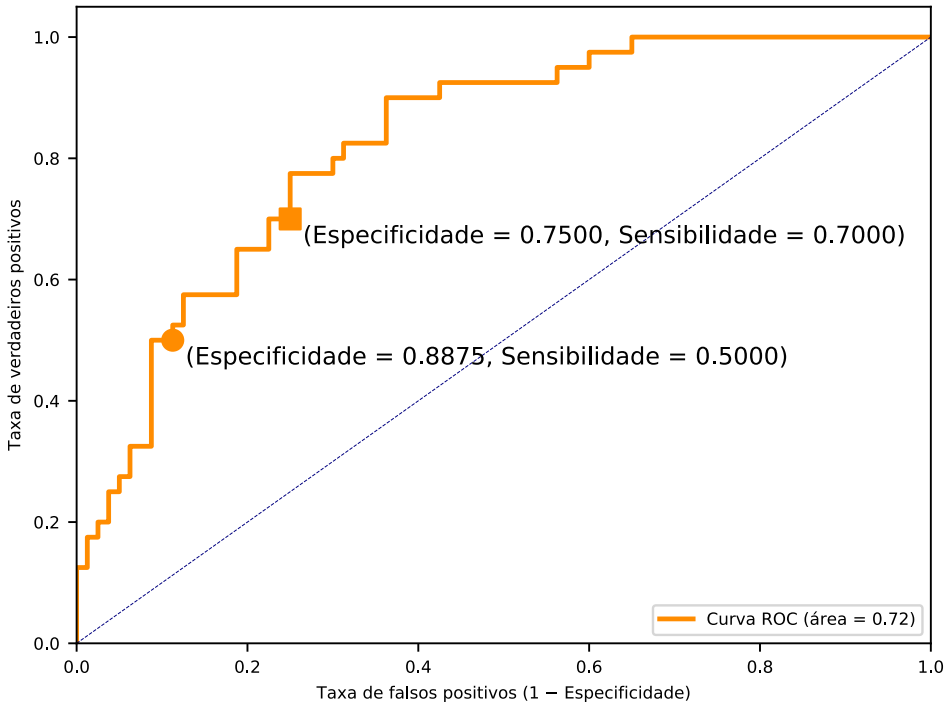

Figura 2.1. Curva ROC (*receiver operator characteristic*) para um modelo de classificação genérico que mostra a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos para um intervalo de limiares de probabilidades de classificação. Dois pontos distintos são destacados e mostram a especificidade e a sensibilidade em dois limiares distintos. A área sob a curva (AUC) possui o valor de $0,72$.

classificação. Essa medida é interessante pois pode sumarizar o desempenho da ROC para um valor escalar único que permite comparar diferentes classificadores. Em outras palavras, a AUC indica o quanto um modelo é capaz de distinguir entre as classes avaliadas pelo modelo [Fawcett, 2006]. Na Figura 2.1, a área sob a curva apresentada possui o valor de $0,72$.

Intuitivamente, um modelo teoricamente perfeito que separa completamente as duas classes teria 100% de sensibilidade e especificidade. Graficamente, a curva ROC desse modelo seria uma única linha do gráfico entre $(0,0)$ e $(0,1)$ e outra linha de $(0,1)$ a $(1,1)$, e a área sob a curva ROC (AUC) seria 1. Um modelo completamente ineficiente resultaria em uma curva ROC seguindo rigorosamente a linha diagonal de 45°, e teria uma área sob a curva (AUC) de aproximadamente 0.5. Para uma comparação visual entre diferentes modelos, pode-se plotar as curvas ROC de cada modelo sobrepostas num mesmo gráfico.

A comparação das curvas ROC pode ser útil para contrastar dois ou mais modelos com diferentes conjuntos atributos (*features*), diferentes parâmetros de treinamento ou

mesmo classificadores completamente distintos. Numa comparação de curvas ROC, o melhor modelo seria aquele mais deslocado ao canto superior esquerdo do gráfico. Além disso, pode-se considerar a maior área sob a curva dos modelos para se determinar o mais eficiente [Kuhn & Johnson, 2013].

As curvas ROC são definidas para problemas de classificação binária, mas existem extensões da técnica que consideram o problema da classificação multiclasse, como as abordagens que calculam a média considerando pesos iguais para a classificação de cada rótulo (*macro-average*) ou a média que considera a contribuição individual de cada classe (*micro-average*) [Kuhn & Johnson, 2013; Fawcett, 2006].

2.5 Teoria de Resposta ao Item (IRT)

A Teoria de Resposta ao Item, ou *Item Response Theory* (IRT), é uma metodologia da área de psicometria que estuda os conceitos por trás de testes, questionários ou instrumentos semelhantes, envolvendo a medição de habilidades e atitudes de um indivíduo, considerando as características dos itens desse questionário [van der Linden & Hambleton, 1997]. Em abordagens IRT, existe uma variável latente de interesse, geralmente baseada em algo compreendido intuitivamente, como "inteligência" ou uma habilidade implícita. Ao contrário de atributos facilmente caracterizados como pesos, alturas, notas, entre outros, a variável latente não pode ser medida diretamente. O objetivo principal das abordagens de psicometria em que se baseia o IRT é a determinação da quantidade ou intensidade de um traço latente que uma pessoa possui. Geralmente, o termo "habilidade" é utilizado para designar esses traços latentes [Andrade et al., 2000].

Para que se possa mensurar a habilidade de um indivíduo, é necessário utilizar uma escala de medidas. Em IRT, qualquer que seja a habilidade de interesse, é definida uma escala cujo ponto médio é zero e os valores de habilidade variam de infinito negativo para infinito positivo. A ideia é que essa escala de medidas pode ser utilizada para dizer o quão habilidosa é uma pessoa e também comparar as habilidades de diferentes pessoas. Para se obter as medidas de habilidade, desenvolve-se um teste consistindo de um certo número de questões. Essas questões são denominadas itens. Idealmente, cada item deve medir um aspecto da habilidade de interesse. A ideia desta abordagem é que cada examinado responde a um item do questionário com base em um nível de habilidade inerente [Baker, 2001].

2.5.1 Curva de característica de item (ICC)

Pode-se considerar que, ao responder o questionário, cada examinado tem um valor numérico de pontuação (*score*) que o posiciona em algum lugar na escala de habilidade. A escala de habilidade é denotada por θ. A cada nível de habilidade na escala, haverá uma certa probabilidade de que um examinado com esta habilidade específica dará uma resposta correta ao item, e esta probabilidade é denotada por $P(\theta)$ [Baker & Kim, 2017]. Em geral, essa probabilidade é baixa para indivíduos com baixa habilidade e alta para indivíduos com alta habilidade. A função $P(\theta)$, quando plotada, tem um formato geral semelhante às curvas apresentadas na Figura 2.2, que ilustra três curvas que descrevem o relacionamento entre a probabilidade de resposta correta a um item e a escala de habilidade. Em IRT, cada curva é associada a um item e é denominada curva de característica de item (*Item Characteristic Curve* - ICC) [Baker, 2001].

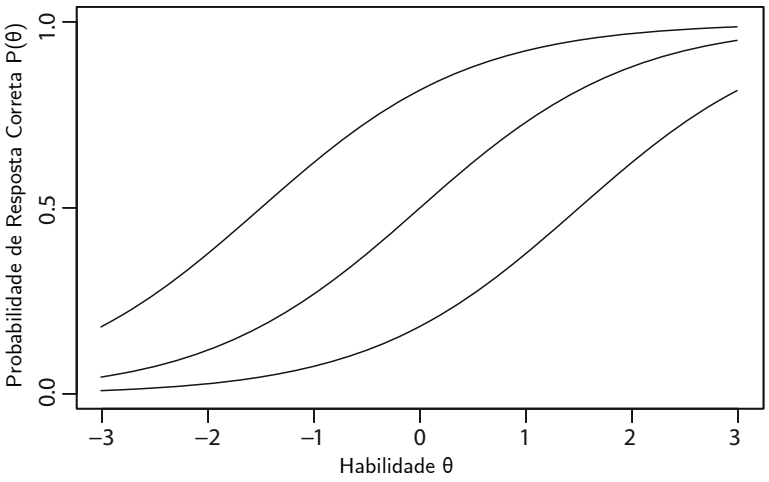

Figura 2.2. Três curvas de características de item com diferentes níveis de dificuldade. Fonte: adaptado de Baker & Kim [2017].

A ICC oferece duas propriedades importantes que são utilizadas para descrever a curva. A primeira é a noção de dificuldade do item, que descreve em que ponto da escala de habilidade o item se encontra. Por exemplo, um item fácil oferece probabilidades de resposta correta maiores para examinados com habilidades mais baixas, enquanto um item difícil apresenta maiores probabilidades de acerto para examinados com habilidades mais altas. A segunda propriedade é denominada discriminação e descreve quão bem um item pode diferenciar entre examinados de baixa ou alta habilidade. Esta propriedade reflete basicamente na inclinação da curva de característica do item. Quanto maior a inclinação da curva, melhor um item pode diferenciar examinados. Quanto mais achatada a curva, menos um item consegue discriminar, já que

a probabilidade de respostas corretas em níveis mais baixos de habilidade serão muito próximos dos níveis mais altos de habilidade [Baker & Kim, 2017].

A ideia de dificuldade de itens pode ser visualizada ainda na Figura 2.2: as três curvas ilustradas apresentam níveis de dificuldades distintos. A curva mais à esquerda representa um item fácil, pois a probabilidade de resposta correta é alta para examinados com baixas habilidades e próxima de 1 para examinados mais habilidosos. A curva central representa um item com dificuldade intermediária, já que a probabilidade de resposta correta é baixa para examinados com níveis de habilidade mais baixos, por volta de 0.5 na região intermediária da escala de habilidades e a probabilidade é próxima de 1 para indivíduos com as mais altas habilidades. Por fim, a curva mais à direita representa um item difícil: a probabilidade de resposta correta é baixa para a maior parte da escala de habilidades, atingindo valores mais altos apenas em níveis de habilidade mais altos. Mesmo no nível de habilidade mais alto ilustrado (+3), a probabilidade de resposta correta é de aproximadamente 0.8 [Baker & Kim, 2017].

O conceito de discriminação de itens é apresentado na Figura 2.3, que contém três curvas de característica de itens com o mesmo nível médio de dificuldade, mas com níveis de discriminação distintos. A curva mais acima tem um alto nível de discriminação, já que a curva é mais íngreme, indicando que a probabilidade de resposta correta muda rapidamente conforme o nível de habilidade aumenta. A curva intermediária representa um item com um nível moderado de discriminação. A inclinação da curva é menor e a probabilidade de resposta correta muda menos rapidamente em relação à curva anterior. Por fim, a terceira curva representa um item com baixa discriminação. Sua inclinação é muito baixa e a probabilidade de resposta correta muda muito lentamente sobre a escala de habilidades. Nesta curva, mesmo em níveis baixos de habilidade, a probabilidade de resposta correta é relativamente alta, o que significa que o item não consegue distinguir bem um indivíduo menos habilidoso de um mais habilidoso [Baker & Kim, 2017].

Um fenômeno comum na condução de testes é que indivíduos examinados possam acertar itens através de um "chute". Este evento pode ser caracterizado em IRT através da utilização de um terceiro parâmetro, adivinhação (*guessing*), além da dificuldade e discriminação [Baker & Kim, 2017].

2.5.2 Modelos de ICC

A ICC fornece visualmente uma noção geral da dificuldade e discriminação de itens, mas seus valores concretos são calculados por meio de modelos matemáticos. Esses modelos oferecem uma equação que relaciona a probabilidade de resposta correta com

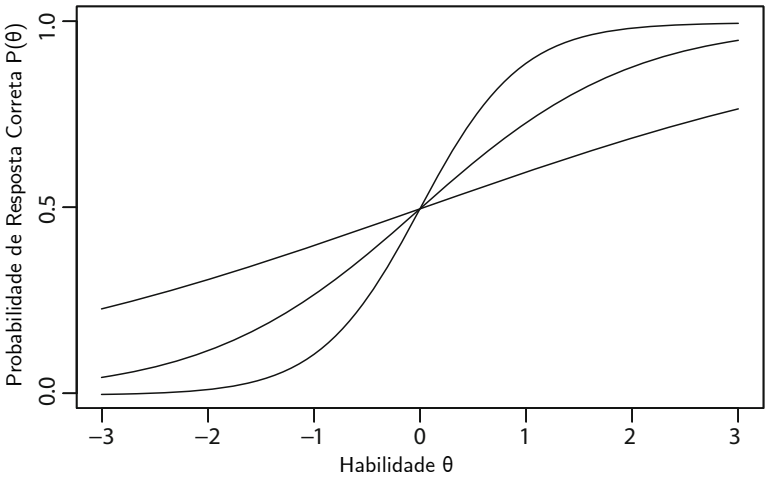

Figura 2.3. Três curvas de características de item com diferentes níveis de discriminação. Fonte: adaptado de Baker & Kim [2017].

a habilidade. Em IRT, o modelo matemático para a curva de característica de item é uma forma cumulativa da função logística [Baker & Kim, 2017]. Existem basicamente três modelos principais que descrevem as curvas e que diferenciam pelo número de parâmetros do item que são estimados. Esses modelos são chamados de modelos logísticos de um, dois e três parâmetros. O modelo mais completo é o de três parâmetros e também é conhecido por 3PL (*3-parameter logistic model*). O modelo 3PL é descrito pela equação 2.1 a seguir.

$$P(\theta) = c + (1 - c)\frac{1}{1 + e^{-a(\theta - b)}} \tag{2.1}$$

onde

θ é o nível de habilidade do indivíduo;

b é o parâmetro dificuldade do item e indica a posição da curva logística;

a é o parâmetro discriminação, ou quanto o item diferencia indivíduos;

c é o parâmetro adivinhação, representando a probabilidade de um acerto casual.

O modelo de dois parâmetros (2PL) pode ser derivado da equação 2.1, simplificando o modelo ao desconsiderar o parâmetro adivinhação (*guessing*), isto é, definindo $c = 0$. Além disso, o modelo de um parâmetro (1PL, ou ainda modelo Rasch) pode ser definido ao desconsiderar o parâmetro discriminação, isto é, admitindo $a = 1$.

Como os valores reais dos parâmetros de itens em um questionário são desconhecidos, uma das tarefas realizadas em IRT é a estimativa desses parâmetros. O método comumente utilizado é a estimativa de máxima verossimilhança (*Maximum-Likelihood Estimation* - MLE) dos parâmetros dos itens. Neste método, o objetivo é encontrar

uma curva de característica de item que melhor se encaixa nas observações de respostas corretas de indivíduos. Para isso, deve-se utilizar um dos modelos descritos anteriormente para se adequar a curva. A Figura 2.4 apresenta uma curva encaixada considerando as proporções de respostas corretas, plotadas como círculos no gráfico.

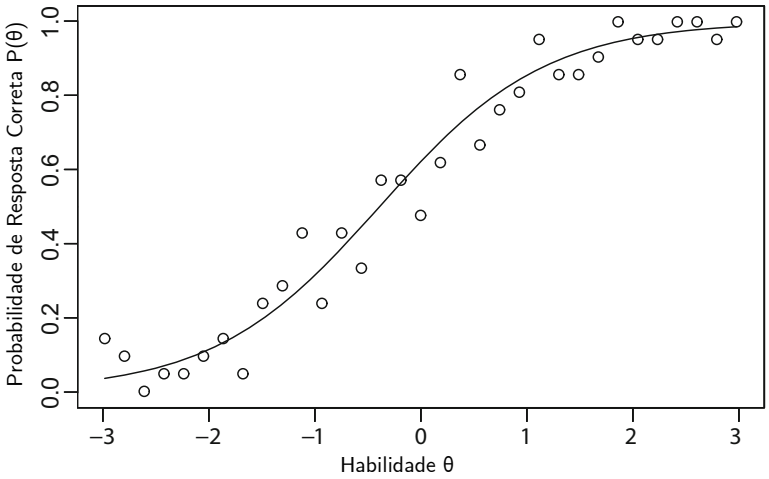

Figura 2.4. Curva de característica de item encaixada nas proporções de respostas corretas de indivíduos. Fonte: adaptado de Baker & Kim [2017].

Os conceitos de IRT apresentados até então são focados em cada item de um questionário. A teoria oferece métodos que avaliam todos os itens de uma só vez, focando as análises no questionário como um todo. O valor de *true score*, calculado pela equação 2.2 a seguir, calcula uma pontuação do questionário através da resposta dada pelos examinados para cada item, considerando a modelagem de ICC empregada (1PL, 2PL ou 3PL). A equação permite calcular o valor *true score* para indivíduos que tenham um dado nível de habilidade (θ).

$$TS_i = \sum_{j=1}^{J} P_j(\theta_i) \tag{2.2}$$

onde

TS_i é o *true score* para examinados com nível de habilidade θ;

j denota um item, $j = 1, ..., J$;

$P_j(\theta_i)$ depende do modelo de ICC empregado.

O cálculo de *true score* pode ser para qualquer ponto na escala de habilidade, de infinito negativo até infinito positivo, formando uma pontuação do questionário como um todo. Essa pontuação geral pode oferecer uma visão relativa a diversos indivíduos examinados com habilidades distintas e formar uma curva semelhante à ICC, que pode

ser interpretada da mesma forma: níveis de habilidade mais baixos possuem menor probabilidade de acerto de itens, enquanto habilidades mais altas possuem maiores probabilidades de acerto de itens.

Capítulo 3

Aprendizado de máquina para análise de leite

Neste capítulo são apresentadas abordagens para a detecção de adulteração do leite bovino com a utilização de diferentes métodos de aprendizado de máquina. Inicialmente, destaca-se a importância da avaliação de qualidade do leite, bem como apresenta-se a espectroscopia no infravermelho, uma técnica capaz de analisar a composição de amostras do leite e de outros materiais. Em seguida, apresenta-se o estudo onde milhares de amostras de leite foram coletadas e submetidas à espectroscopia no infravermelho. Os dados produzidos pela técnica utilizada foram analisados e utilizados como entrada para experimentos com diferentes métodos, que incluem Regressão Logística, Partial Least Squares, Random Forest e Redes Neurais. Dentre os métodos, apresenta-se a proposta de uma nova arquitetura de Rede Neural Convolucional para caracterização de regiões espectrais do leite, que foi aplicada à detecção de adulteração e apresenta desempenho superior quando comparada a outros métodos. Diferentes experimentos para avaliação dos métodos foram realizados e os detalhes das avaliações são apresentados. Os resultados mostram que pode-se obter uma precisão na detecção de adulterantes de até 98,76%.

3.1 Qualidade do leite

O leite bovino é um dos alimentos mais importantes e um dos mais consumidos pela humanidade. Sua importância é reforçada por seu valor nutricional e suas diversas aplicações na indústria de produtos para consumo. Por isso, garantir a qualidade do produto para comercialização e consumo humano é extremamente importante.

A qualidade do leite pode ser deteriorada por práticas de adulterações fraudulentas, que consistem em adicionar substâncias externas ao leite com diferentes objetivos. Essa é uma prática comum no Brasil e em vários países pelo mundo [Kartheek et al., 2011], com os objetivos mais comuns sendo o aumento do volume do produto, a camuflagem de parâmetros de qualidade não conformes e o aumento do lucro com práticas ilegais [Alves da Rocha et al., 2015; de Carvalho et al., 2015; Santos et al., 2013]. Diferentes substâncias são adicionadas ao leite com propósitos específicos, como a sacarose e o amido, que são usados para modificar a densidade do produto e o ponto de congelamento após ser modificado com acréscimo de água. Bicarbonato de sódio é uma substância que pode reduzir altos níveis de acidez relacionados à contaminação bacteriana, associados a más práticas de fabricação. Peróxido de hidrogênio e formaldeído podem preservar a contagem microbiana relacionada à má qualidade do leite [Gondim et al., 2017].

Os testes qualitativos clássicos para detectar adulteração no leite, estabelecidos como métodos oficiais pela autoridade reguladora brasileira, são métodos de bancada que exigem uma grande quantidade de testes e reagentes, consomem tempo e geram grandes quantidades de resíduos. Alternativamente, há técnicas instrumentais não destrutivas, que consomem menos reagentes, geram menos resíduos e economizam tempo e recursos [Botelho et al., 2015; Souza et al., 2011]. Uma das técnicas mais utilizadas na indústria alimentar é a espectroscopia no infravermelho, cujas vantagens incluem a análise de amostras com pouca ou nenhuma preparação, a facilidade de uso e rápida obtenção de dados considerados "impressão digital" de amostras [Gondim et al., 2017].

3.2 Espectroscopia no infravermelho

A espectroscopia no infravermelho é uma técnica da área da Quimiometria que mede o comprimento de onda e a intensidade da luz infravermelha absorvida por um material. A Quimiometria é a ciência que estuda a aplicação de métodos estatísticos e computacionais em dados de origem química [Santos et al., 2013]. O espectro infravermelho de uma amostra é registrado pela passagem de um feixe de luz infravermelha considerando as vibrações dos átomos, posteriormente determinando-se a absorção da radiação em comprimentos de onda específicos. Esta técnica permite determinar grupos funcionais presentes em uma amostra, uma vez que cada grupo funcional de uma molécula possui uma frequência vibracional única. O espectro final registrado representa uma configuração molecular única da composição da amostra analisada [Kamal & Karoui, 2015].

O equipamento de espectroscopia emite diversos comprimentos de onda que passam pelo material e que são, posteriormente, registrados a partir de um conjunto de diferentes posições de um espelho. Os valores registrados por este processo devem ser manipulados com o objetivo de transformar os dados que representam a absorção da luz em diversas posições do espelho nos dados interpretáveis, que representam a absorção da luz para cada comprimento de onda. Esse processo é obtido pela aplicação da Transformada de Fourier [Griffiths et al., 2007]. A técnica é, portanto, conhecida como *Fourier-transform infrared spectroscopy* (FTIR), ou Espectroscopia no Infravermelho com Transformada de Fourier.

O infravermelho pode ser dividido em três regiões espectrais: o infravermelho próximo de alta energia (NIR) (≈14.000 e 4.000 cm^{-1}), o infravermelho médio (MIR) (≈4.000 e 400 cm^{-1}) e o infravermelho distante (≈400 e 10 cm^{-1}). Em uma leitura de um espectro, cada molécula apresentará o seu próprio espectro na região do infravermelho, tornando o método viável para identificar diferentes tipos de amostras. Os picos presentes no gráfico do espectro correspondem às frequências de vibrações entre os átomos que compõem cada amostra. A altura desses picos corresponde à quantidade de determinada amostra [Kamal & Karoui, 2015].

Na indústria de alimentos, o FTIR é uma das técnicas mais utilizadas para a leitura de composição de amostras. FTIR oferece uma metodologia simples, rápida e não destrutiva que pode ser aplicada à análise de composição do leite [Santos et al., 2013]. Os dados espectrais gerados podem ser explorados computacionalmente, o que torna a técnica de grande interesse para a Ciência da Computação.

3.3 Aquisição de dados e preparação das amostras

Amostras de leite foram coletadas e analisadas pelo Laboratório de Análise da Qualidade do Leite da Escola de Veterinária da Universidade Federal de Minas Gerais (EV-UFMG), credenciado pelo Ministério da Agricultura, Pecuária e Abastecimento (MAPA) e certificado pelas normas ABNT NBR ISO/IEC 17025:2017. Duas origens distintas compõem as amostras utilizadas para o presente estudo: amostras provenientes da fazenda experimental da EV-UFMG, localizada no município de Igarapé, MG, e amostras provenientes de análises comerciais realizadas pelo laboratório em seus procedimentos de rotina. Um total de 4.846 amostras foram coletadas e 2.376 amostras foram adulteradas de forma controlada para a realização de experimentos deste trabalho.

A cada amostra de adulteração foi adicionada uma substância com concentrações

específicas para simular as características de fraudes na indústria mais comumente praticada no país [Santos et al., 2013; Botelho et al., 2015]. As substâncias utilizadas como adulterantes foram: sacarose, amido, bicarbonato de sódio, peróxido de hidrogênio e formaldeído. Os adulterantes foram diluídos no leite de forma padronizada e posteriormente submetidos ao equipamento de FTIR (LactoScope™ FTIR 400, Delta Instruments, Drachten, Holanda), para a obtenção das leituras espectrais.

Para cada amostra analisada, o equipamento FTIR gera dois arquivos com dados distintos: um arquivo de espectro infravermelho, no formato SPC, que contém as coordenadas do espectro lido, e um arquivo de componentes, no formato CSV, que contém variáveis numéricas, chamadas de *component features* (atributos de componentes), calculadas pelo equipamento a partir do espectro. As informações provenientes de ambas as fontes (arquivos SPC e CSV) foram analisadas e condensadas em um único conjunto de dados no formato CSV, a fim de facilitar a manipulação de informações. Este conjunto de dados, contém, portanto, para cada amostra, as coordenadas espectrais da leitura de equipamento, bem como as variáveis numéricas referentes aos componentes do leite. Além disso, o conjunto de dados contém, para cada instância, o atributo *Ingredient*, que representa o adulterante acrescentado à amostra, isto é, o rótulo de classe (*class label*) da amostra.

3.3.1 Análise de componentes

Durante o processo de leitura do espectro infravermelho, o equipamento FTIR realiza uma série de cálculos que determinam valores numéricos para diferentes componentes do leite. Esses cálculos são realizados a fim de obter uma estimativa da qualidade da amostra com base em características amplamente utilizadas pelos laboratórios de análise. Um total de 10 variáveis são obtidas e dependem de calibrações no equipamento que, a partir de um modelo de regressão linear múltipla (*Multiple Linear Regression*, MLR) interno ao equipamento, considera a absorção da luz por regiões pré-determinadas. O modelo de regressão é capaz de estimar numericamente as seguintes *features*: gordura (fat), proteína (*protein*), lactose, sólidos totais (*total solids*), sólidos não-gordurosos (*solids non-fat*, SNF), caseína (*casein*) e nitrogênio ureico do leite (*milk urea nitrogen*, MUN). Outros três valores também são calculados: a contagem de células somáticas (*Cells*), o valor do ponto de congelamento (*FrzPoint*) e um valor de controle de qualidade da amostra (*QValue*). Os atributos componentes, juntamente com suas unidades de medida, de um subconjunto aleatório de amostras são apresentados na Figura 3.1 (a).

A fim de entender o relacionamento entre as variáveis, correlações par-a-par foram

(a)

Cells (no. cells/µL)	QValue (%)	Fat (g/100g)	Protein (g/100g)	Lactose (g/100g)	Solids (g/100g)	FrzPoint (°H)	SNF (g/100g)	MUN (mg/dL)	Casein (g/100g)	Ingredient
59	95.10	3.42	3.38	4.73	12.50	0.578	9.14	11.1	2.62	pure
58	90.74	3.42	3.39	4.77	12.55	0.582	9.20	13.0	2.63	formaldehyde
46	91.76	3.37	3.35	4.74	12.44	0.578	9.12	27.8	2.56	peroxide
74	91.54	3.5	3.37	4.85	12.67	0.591	9.22	16.1	2.66	sucrose
707	90.83	3.66	3.33	4.80	12.69	0.558	9.35	10.7	2.61	starch
291	85.76	4.13	3.37	4.72	13.19	0.554	9.37	14.8	2.66	bicarbonate

(b)

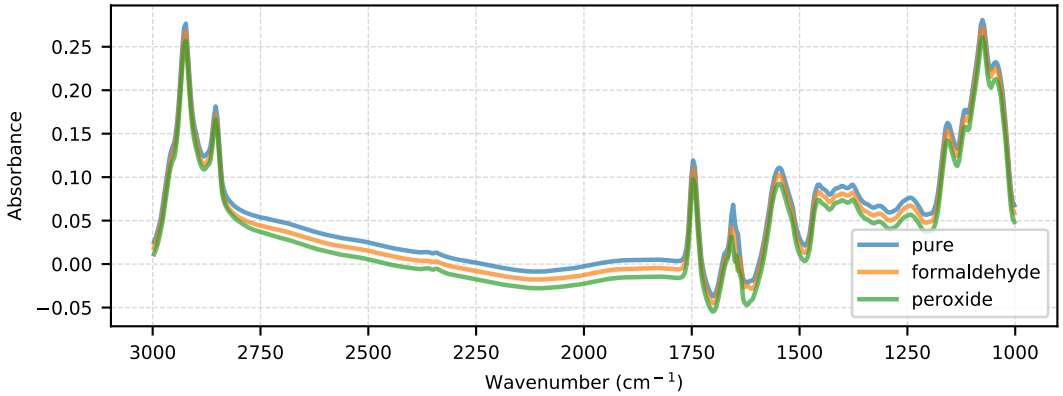

Figura 3.1. (a) Atributos componentes para um subconjunto aleatório de amostras do leite gerado pelo equipamento FTIR após a leitura dos espectros. Cada coluna quantifica uma informação de composição importante do leite. As colunas *fat*, *protein*, *lactose*, *solids*, SNF, *casein* e MUN são apresentadas em unidades de concentração. A coluna Cells representa a contagem de células somáticas. FrzPoint representa o ponto de congelamento do material, com valores dados em graus Hortvet (°H) e QValue é um cálculo da qualidade da amostra realizado pelo equipamento. (b) Plotagem do espectro infravermelho para três amostras selecionadas aleatoriamente com os rótulos de classe puro, formol e peróxido. Cada espectro foi plotado com um pequeno deslocamento para evitar sobreposição e facilitar a visualização.

calculadas a partir da normalização das variáveis componentes do conjunto de dados. O relacionamento das variáveis demonstram que proteína e caseína são altamente correlacionadas (0.96). Como a caseína é uma proteína do leite, a correlação faz sentido. Outras correlações podem ser destacadas, como sólidos e gordura (0.85), lactose com ponto de congelamento (0.77) e lactose com sólidos não gordurosos (0.81). Outros pares de variáveis não apresentaram correlações significativas. A correlação completa é apresentada na Figura 3.2.

Os atributos componentes apresentam uma grande variância e escalas diferentes. Como essa característica pode impactar na execução de métodos estatísticos e de aprendizado de máquina, uma análise de variância foi realizada. A Figura 3.3 apresenta

um *boxplot* considerando a escala e a variância de todos os atributos componentes.

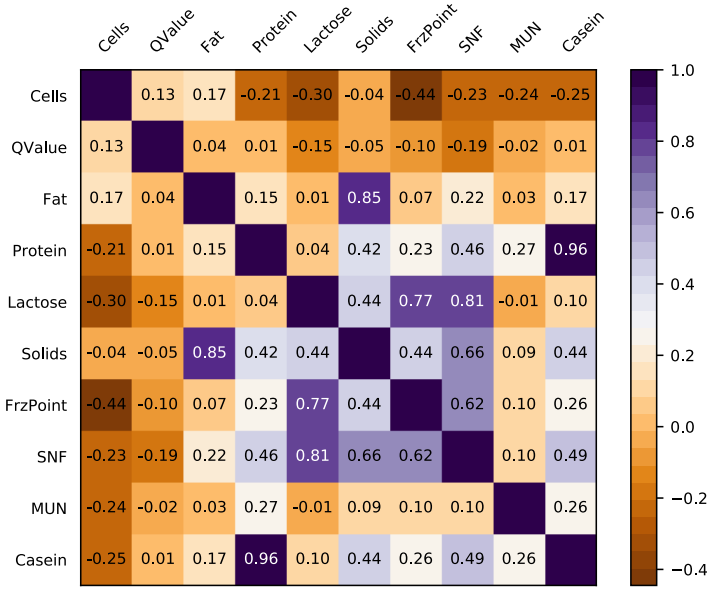

Figura 3.2. Matriz de correlação de atributos componentes. Os valores foram calculados utilizando o coeficiente de correlação de Pearson a partir da normalização das variáveis do conjunto de dados. Os valores indicam que caseína e proteína são altamente correlacionadas, o que é esperado, já que a caseína é uma proteína. Outras correlações são significativas, como sólidos com gordura, lactose com ponto de congelamento e lactose com sólidos não gordurosos. Outras variáveis não são significativamente correlacionadas.

3.3.2 Análise de espectros infravermelhos

O espectro infravermelho é a principal informação que o equipamento FTIR gera durante a análise de cada amostra. É a partir do espectro em si que o equipamento executa um método de regressão para consolidar os dez atributos componentes descritos anteriormente. O espectro infravermelho é muito rico em informações, mas é necessário que se realize determinados processamentos com os dados para que informações úteis possam ser obtidas para quaisquer tomadas de decisão. Um exemplo de processamento é o cálculo que o próprio equipamento realiza através do modelo de regressão proprietário do fabricante, a partir de um amplo conhecimento do leite e de calibrações realizadas previamente. No entanto, a simples representação da composição de uma amostra em 10 atributos pode desperdiçar dados importantes. Torna-se, então, importante o estudo mais aprofundado dos dados espectrais puros determinados pelo processo de espectroscopia.

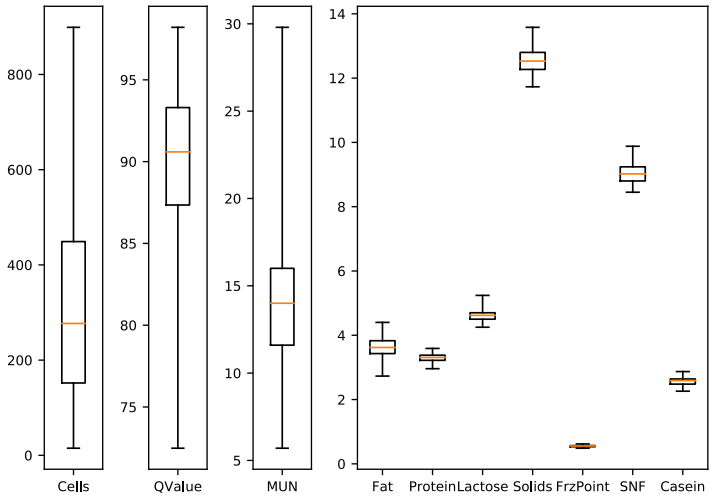

Figura 3.3. *Boxplot* dos atributos componentes. A plotagem mostra escalas e variâncias de cada atributo. Cells, QValue e MUN possuem escalas significativamente diferentes do restante dos atributos e, portanto, foram plotadas separadamente. O atributo Cells mostra que as amostras possuem uma contagem de células somáticas de ≈150 a ≈450 células/μL. O valor de QValue mostra que as amostras são em geral consideradas de alta qualidade (≈87% a ≈93%). Todos os outros atributos apresentam pouca variância.

Para cada amostra no conjunto de dados, tem-se as coordenadas do espectro lidas durante a análise do equipamento. Essas coordenadas são compostas de 518 pontos e representam a absorbância de cada números de onda (ou região espectral). As coordenadas espectrais partem de 3.000 cm^{-1} até 1.000 cm^{-1}, medidos em número de onda. A Figura 3.1 (b) apresenta espectros de três amostras selecionadas aleatoriamente do conjunto de dados, juntamente com seus rótulos de classe.

3.4 Execução de métodos de aprendizado de máquina

Para a avaliação dos métodos de aprendizado de máquina propostos neste estudo, dois problemas em relação à adulteração do leite foram considerados: o problema de classificação multiclasse e o problema de classificação binária. Para a classificação multiclasse, o atributo *Ingrediente* (rótulo de classe) das amostras pode assumir valores relativos a um dos adulterantes utilizados no estudo (bicarbonato, formaldeído, peróxido de hidrogênio, amido ou sacarose), ou o valor *leite cru*, quando não houve adulterante utilizado e trata-se de uma amostra pura. Para a classificação binária, o rótulo de classe *Ingrediente* do conjunto de dados foi transformado, considerando-se a presença ou ausência de adulterantes, com valor *leite cru* para amostras puras e *leite adulterado* para amostras adulteradas, independente do ingrediente adulterante

considerado.

As estruturas dos dados relativas aos atributos componentes (10 *features*) e às coordenadas espectrais (518 pontos) exigem diferentes abordagens na aplicação de métodos de aprendizado de máquina. Nos atributos componentes, cada *feature* representa fortemente uma característica da composição do leite, enquanto as coordenadas espectrais representam curvas do espectro infravermelho, onde cada valor é um ponto da curva. Nos dois casos, métodos diferentes de aprendizado de máquina podem ser aplicados. Nos experimentos realizados, para os dados dos atributos componentes, árvores de decisão em métodos *ensemble* foram aplicadas, mais especificamente, Random Forest e Gradient Boosted Trees. Como métodos *ensemble* realizam a combinação de modelos para a solução de um problema, eles aumentam o desempenho preditivo quando comparados a modelos individuais [Dong et al., 2020]. Para os dados espectrais, Redes Neurais Convolucionais (CNN) foram utilizadas. Pode-se entender que a CNN interpreta o espectro da amostra do leite como uma "imagem", tal como se propõem as camadas convolucionais de detecção de filtros que selecionam características importantes do espectro.

3.4.1 Seleção de conjuntos de dados

A fim de estimar a qualidade dos modelos utilizados para a tarefa de classificação nos experimentos, o conjunto de dados inicial, contendo 4.846 amostras, foi subdividido em conjuntos de treinamento e de teste. Três pares de conjuntos de treinamento e teste foram obtidos: o primeiro contendo 90% das instâncias para treinamento e 10% dos dados para teste; o segundo com 75% para treinamento e 25% para teste; e o terceiro, com 50% para treinamento e 50% para teste. Cada seleção de um dos pares treinamento/teste foi obtida do conjunto de dados inicial aleatorizado. Essa estratégia de divisão do conjunto de dados, denominada *hold-out*, é uma das variações de *cross-validation* e está descrita no Capítulo 2, Seção 2.4.2. A Tabela 3.1 apresenta a distribuição de instâncias de cada separação do conjunto de dados considerando a classificação multiclasse. Na versão de classificação binária, apresentada na Tabela 3.2, as cinco classes referentes a substâncias adulterantes são somadas em uma única classe denominada *adulterated*.

3.4.2 Métodos de benchmark

Para que pudessem ser definidos os métodos propostos para a análise de adulteração no leite baseados em árvores *ensemble* e *deep learning*, e para que houvesse um parâmetro

Tabela 3.1. Distribuição de classes para as instâncias em cada separação de treinamento e teste em leite cru ou adicionado com diversos adulterantes.

Separação do conjunto de dados	Classes	Número de instâncias de treinamento	Número de instâncias de teste
90/10%	Leite cru	2213	257
	Leite cru + bicarbonato	419	41
	Leite cru + formaldeído	442	43
	Leite cru + peróxido de hidrogênio	417	48
	Leite cru + amido	439	41
	Leite cru + sacarose	431	55
75/25%	Leite cru	1846	624
	Leite cru + bicarbonato	338	122
	Leite cru + formaldeído	347	138
	Leite cru + peróxido de hidrogênio	364	101
	Leite cru + amido	359	121
	Leite cru + sacarose	380	106
50/50%	Leite cru	1239	1231
	Leite cru + bicarbonato	219	241
	Leite cru + formaldeído	223	262
	Leite cru + peróxido de hidrogênio	242	223
	Leite cru + amido	253	227
	Leite cru + sacarose	247	239

Tabela 3.2. Distribuição de classes para as instâncias em cada separação de treinamento e teste considerando o problema de classificação binária.

Separação do conjunto de dados	Classes	Número de instâncias de treinamento	Número de instâncias de teste
90/10%	Leite cru	2213	257
	Leite adulterado	2148	228
75/25%	Leite cru	1846	624
	Leite adulterado	1788	588
50/50%	Leite cru	1239	1231
	Leite adulterado	1184	1192

de desempenho dos modelos, executou-se inicialmente métodos estatísticos mais simples e comumente utilizados para análises do leite. Estes métodos são importantes para definir um *benchmark* para os modelos propostos no experimento. Os métodos de Regressão Linear e Regressão Logística são considerados estatísticos simples quando comparados a outros métodos de aprendizado de máquina. O primeiro utiliza uma função linear para modelar o relacionamento entre variáveis dependentes e variáveis independentes. O segundo utiliza a função logística para modelar a relação entre as variáveis. Já o Partial Least Squares (PLS) é um método muito comum para a avaliação da qualidade e da autenticidade do leite e produtos derivados [Santos et al., 2013; Botelho et al., 2015; Kasemsumran et al., 2007; Nicolaou et al., 2010; Luna et al.,

2016]. Todos os modelos foram utilizados em suas implementações padrão da biblioteca Scikit-learn [Pedregosa et al., 2011]. Os algoritmos foram implementados considerando-se o problema de classificação, a fim de oferecer comparações de métricas equivalentes com o restante dos modelos implementados no estudo.

Os modelos foram treinados e testados considerando os três pares de conjuntos descritos na Seção 3.4.1, deste capítulo. Ambas as versões de problemas binário e multiclasse foram analisadas. Para a Regressão Linear, os valores de acurácia variaram de 31,55% a 33,50% para o problema multiclasse e 79,20% a 79,62% para a classificação binária. Na Regressão Logística, as acurácias variaram de 55,92% a 58,76% na classificação multiclasse e de 71,40% a 76,49% na classificação binária. Por fim, para PLS, as acurácias foram de 32,56% a 35,26% em multiclasse e de 76,91% a 77,39% na classificação binária. Pode-se observar, nos resultados, que, embora todos os métodos obtivessem um desempenho relativamente bom nas versões binárias das classificações, os resultados não foram satisfatórios para as versões multiclasse. Todos os resultados de acurácia para os três métodos avaliados são apresentados na Tabela 3.3.

Tabela 3.3. Resultados da execução de métodos para classificações binária e multiclasse usados como *benchmark* para os modelos avaliados no trabalho. Todos os classificadores foram executados com os três pares de treinamento e teste obtidos aleatoriamente a partir do conjunto de dados inicial.

Conjunto de dados	Versão de classificação	Regressão Logística	Regressão Linear	Partial Least Squares (PLS)
90/10%	Multiclasse	0.5876	0.3155	0.3526
	Binária	0.7649	0.7959	0.7691
75/25%	Multiclasse	0.5693	0.3350	0.3267
	Binária	0.7583	0.7962	0.7739
50/50%	Multiclasse	0.5592	0.3281	0.3256
	Binária	0.7140	0.7920	0.7714

3.4.3 Métodos ensemble

Os métodos *ensemble* considerados neste estudo são métodos baseados em conjuntos de árvores de decisão, mais especificamente Random Forest (RF) e Gradient Boosting Machines (GBM), aplicados aos atributos componentes (*features*) calculados internamente pelo equipamento FTIR através de modelos de regressão proprietários e calibrações do equipamento. Os modelos de árvore são adequados a dados com as características das *features* pois cada valor representa fortemente uma informação de composição do leite

e representa uma característica importante, como concentrações de células somáticas, de gordura, lactose, etc. Os métodos de árvores de decisão mais comuns são ID3, C4.5 e CART [Maimon & Rokach, 2010]. Porém, os métodos *ensemble* utilizam um conjunto de árvores como peças para a construção de modelos com maior poder de predição [James et al., 2017]. Além disso, esses modelos têm capacidades de generalização melhores que modelos individuais, diminuindo as chances de uma predição ruim em um determinado caso específico [Polikar, 2006].

Os modelos de RF e GBM utilizam as implementações disponíveis no Scikit-learn [Pedregosa et al., 2011]. Todos os parâmetros dos algoritmos utilizaram os valores padrão, exceto o parâmetro `n_estimators`, que foi definido como 200. Este parâmetro define o número de preditores utilizados no treinamento de cada modelo *ensemble*. Os modelos foram avaliados com cada conjunto de treinamento e teste, 90%/10%, 75%/25% e 50%/50%, para os problemas de classificação binária e multiclasse.

3.4.4 Método de deep learning

Os dados referentes às coordenadas espectrais produzidos pelo processo de espectroscopia no infravermelho são brutos, de forma que, para a aplicação de algum método de aprendizado de máquina que envolva, por exemplo árvores de decisão, alguma manipulação nos dados, a fim de extrair *features*, se torna necessária. As Redes Neurais Convolucionais (CNN), no entanto, oferecem uma característica que soluciona essa questão: as camadas convolucionais são capazes de aprender, com base no treinamento da rede, um número definido de filtros que destacam diferentes características. Os neurônios das camadas convolucionais respondem a estímulos em diferentes regiões espectrais fazendo com que os filtros, então, passam para as camadas profundas da rede os dados espectrais que a rede considera importante. Por isso, as CNN são capazes de receber como entrada os dados espectrais brutos, sem a necessidade de qualquer passo adicional de preprocessamento, podendo realizar a extração de *features* sem a necessidade de quaisquer interações manuais [Schmidhuber, 2015].

Para a execução do método com *deep learning*, uma arquitetura de Rede Neural Convolucional foi proposta. Esta arquitetura possui uma camada convolucional de uma dimensão, para o recebimento dos dados espectrais, que aprende 32 filtros com tamanho de *kernel* 5. Isso significa que a rede irá passar uma janela deslizante pelos dados 32 vezes e que cada janela tem tamanho 5 para os cálculos de convolução (tamanho de *kernel*). A camada pode, portanto, detectar 32 *features* diferentes diretamente a partir do espectro infravermelho. Essa camada também pode ser chamada de camada detectora de *features*. A arquitetura de CNN proposta foi baseada no trabalho de [Liu

et al., 2017], porém com uma estrutura muito mais simples, necessitando de menos camadas, filtros e neurônios. A Figura 3.4 apresenta os detalhes da arquitetura de CNN proposta.

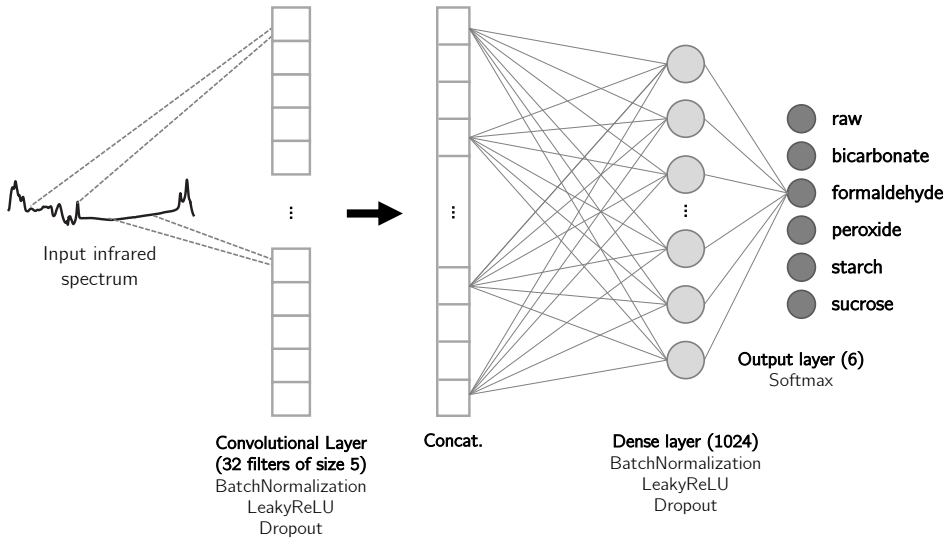

Figura 3.4. Arquitetura da Rede Neural Convolucional (*Convolutional Neural Network*, CNN) para classificação multiclasse de espectros infravermelhos puros. A arquitetura proposta consiste de uma camada convolucional que aprende 32 filtros, com tamanho de *kernel 5*, que reconhece características diretamente do espectro de entrada. A saída da camada convolucional é passada como entrada para uma camada totalmente conectada, com 1024 neurônios. A camada de saída possui tamanho 6, que é o número de classes no problema multiclasse e é ativada pela função *Softmax*.

A saída da camada convolucional é concatenada e passada para uma camada densa, mais conhecida por camada totalmente conectada, composta de 1024 neurônios. São utilizadas camadas auxiliares que realizam a ativação dos neurônios com a função LeakyReLU [Maas et al., 2013], que adicionam não linearidade ao modelo. São realizadas operações de *Batch Normalization* [Ioffe & Szegedy, 2015], que normalizam os dados das camadas e de *Dropout* [Srivastava et al., 2014], que ignoram um determinado número de neurônios aleatoriamente a fim de oferecer um poder maior de generalização do modelo, evitando o efeito de *overfitting* com os dados de treinamento.

Para as classificações binária e multiclasse, uma CNN diferente foi treinada, e elas diferem especificamente no número de neurônios na camada de saída. Como esta camada gera a saída para a classificação, o número de neurônios deve ser exatamente o número de classes possíveis de acordo com os dados. Então, a CNN para a classificação binária possui a última camada com um neurônio com saída binária, ativado pela função sigmoide. A CNN para a classificação multiclasse possui a última camada com

seis neurônios, um para cada classe, ativados pela função *Softmax* [Zeiler & Fergus, 2014]. O modelo binário classifica as amostras entre a presença ou a ausência de um adulterante, enquanto o modelo multiclasse classifica as amostras como leite puro ou uma das cinco substâncias adulterantes conhecidas.

O treinamento da CNN foi feito usando o otimizador Adam [Kingma & Ba, 2014] com a execução de 100 épocas (*epochs*), tanto para os problemas de classificação binária quanto multiclasse. Cada execução da CNN considerou como conjunto de validação 20% do conjunto de treinamento. Durante o treinamento e a validação do modelo, para cada *epoch* da rede, históricos de acurácia e perda são armazenados. A cada *epoch*, a acurácia é calculada comparando-se a classe predita com a classe real, e a perda é calculada pelo valor de entropia cruzada entre a classe predita e a classe real. No cálculo de acurácia, a ideia é, para o problema binário, que as probabilidades de classe próximas a 0 são consideradas como classe *adulterated* e valores próximos a 1 são considerados *pure*. Para o problema multiclasse, a classe predita é a classe com a maior probabilidade dentro das seis possíveis classes. A Figura 3.5 mostra a plotagem da acurácia e da perda durante as *epochs* do modelo. É possível ver que a validação da rede obteve melhores resultados no problema binário, o que é esperado, já que a classificação binária é considerada um problema mais simples que a classificação multiclasse.

A arquitetura da CNN foi implementada utilizando Keras [Chollet et al., 2015] e TensorFlow [Abadi et al., 2016] em Python. Todo o processamento envolvendo as redes foi executado em um computador laptop pessoal, com 8GB de memória e processador Intel Core i5 de 2,7 GHz. O treinamento dos modelos leva aproximadamente 16 minutos e a classificação de todas as amostras nos conjuntos de teste leva apenas 270 milissegundos.

3.5 Extração automática de features

Nos experimentos realizados neste trabalho, os dados utilizados como entrada para a Rede Neural Convolucional são as coordenadas espectrais originais das amostras, sem preprocessamento. Por outro lado, os dados utilizados para os métodos baseados em árvores (RF e GBM) exigem um conjunto de *features* extraídas a partir das coordenadas espectrais, onde cada *feature* representa alguma característica importante das amostras. As *features* utilizadas são provenientes de uma metodologia proprietária do equipamento FTIR que, como descrito anteriormente, na Seção 3.3.1, utiliza regressão linear e calibrações do equipamento para determinar 10 *features* (ou componentes) para

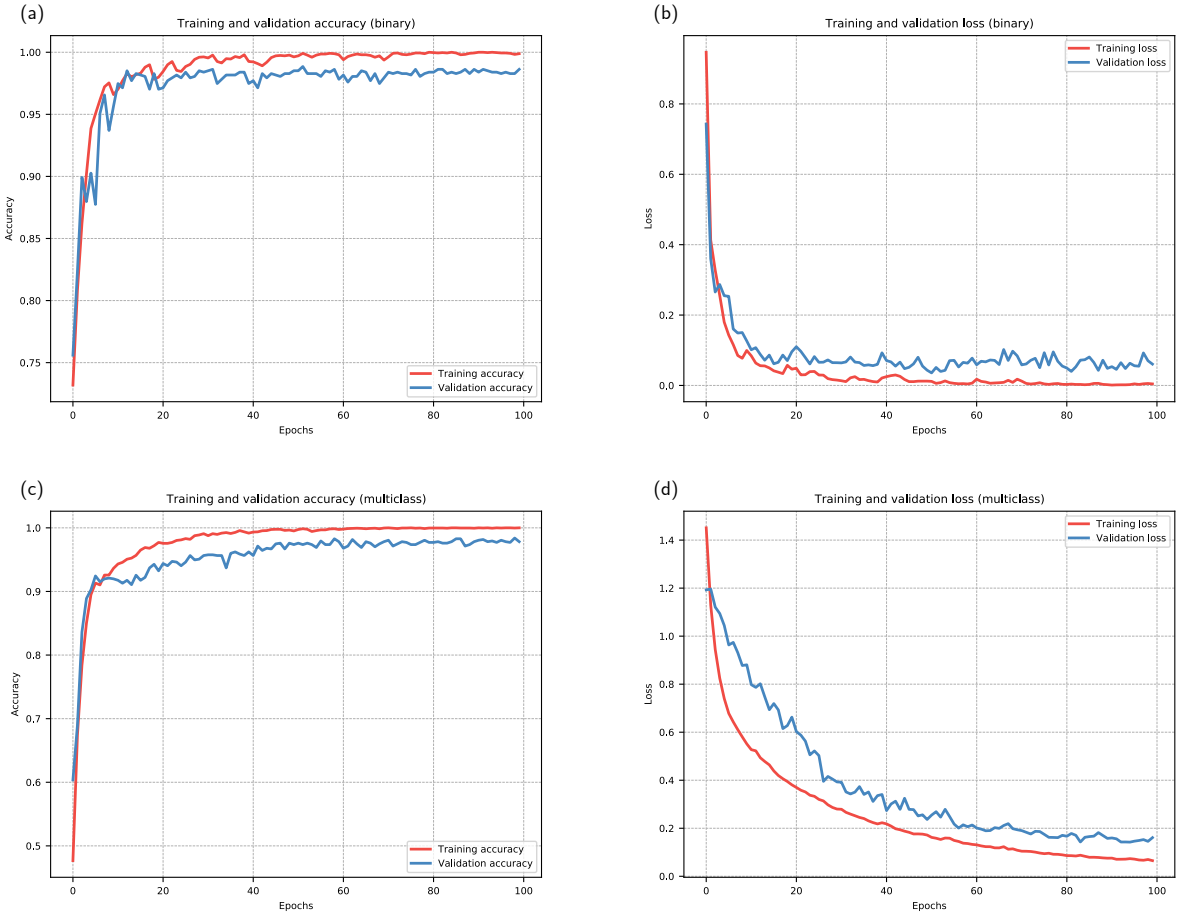

Figura 3.5. Plotagem da acurácia e da perda dos modelos de CNN considerando o par de treinamento e teste 80%/20%. O modelo foi treinado por 100 *epochs*. Acurácia (a) e perda (b) do treinamento e validação considerando o problema binário. Acurácia (c) e perda (d) do treinamento e validação considerando o problema multiclasse.

as amostras analisadas.

A utilização da arquitetura de CNN proposta, além de trazer melhor desempenho que os demais métodos, ainda possibilita a obtenção de uma contribuição adicional. As camadas convolucionais da CNN possuem filtros que aprendem, com os espectros de entrada, as regiões mais significativas e representativas dos espectros. Essas regiões são denominadas mapa saliência [Simonyan et al., 2013]. Como as CNN são aplicadas geralmente para o reconhecimento de imagens, estudos recentes foram capazes de visualizar os filtros convolucionais da rede, considerando uma determinada imagem de entrada para a rede [Simonyan et al., 2013; İmamoğlu et al., 2017]. Em outras palavras, é possível visualizar que regiões da imagem foram mais significativas para a rede. A Figura 3.6 apresenta o mapa de saliência de uma CNN para uma determinada imagem de entrada.

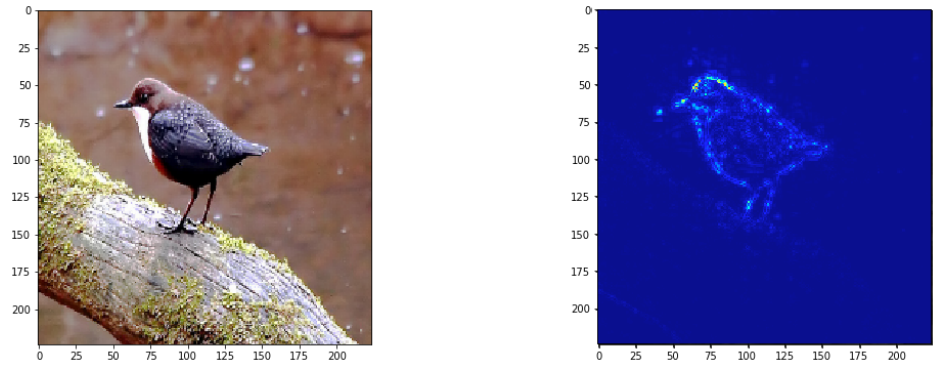

Figura 3.6. Imagem de entrada (à esquerda) e mapa de saliência (à direita) calculado por uma Rede Neural Convolucional (CNN). Fonte: biblioteca para visualização de redes neurais Keras-Vis (`https://github.com/raghakot/keras-vis/`)

Utilizando os conceitos para a determinação do mapa de saliências em imagens, foi aplicada esta metodologia nas coordenadas espectrais obtidas das amostras de leite. Os métodos foram readequados para a diferença de estrutura de entrada, pois, ao contrário de imagens, os espectros obtidos são coordenadas unidimensionais. A obtenção das saliências nos espectros obtidos, realizam, portanto, a extração das regiões espectrais que mais contribuíram para a ativação de uma determinada saída na CNN. A Figura 3.7 apresenta as saliências obtidas com o espectro de uma amostra adulterada com bicarbonato, isto é, as regiões que ativaram o nó final "bicarbonato" na CNN utilizada.

Então, pode-se utilizar os pontos de saliência que ocorrem com mais frequência considerando todas as amostras do conjunto de treinamento para formar um conjunto de regiões espectrais. Essas regiões compreendem os pontos que a rede considerou mais importante utilizando todas as amostras de treinamento. Seleciona-se, assim, os k pontos de saliência mais frequentes, formando-se, portanto, um conjunto de *features* que podem ser utilizadas para substituir o processamento realizado pelo equipamento FTIR. Esse conjunto de *features* é extraído automaticamente, sem conhecimentos prévios das amostras e possui a estrutura necessária para a execução de métodos como os *ensemble* de árvores Random Forest (RF) e Gradient Boosting Machines (GBM).

A Tabela 3.4 apresenta os resultados da classificação com Random Forest utilizando como entrada $k = 60$ *features* mais frequentes obtidas pela técnica de mapa de saliência com a rede convolucional. A Tabela 3.5 apresenta os resultados considerando a execução de Gradient Boosting Machines (GBM) para a mesma técnica.

A extração automática de *features* é importante, uma vez que permite a deter-

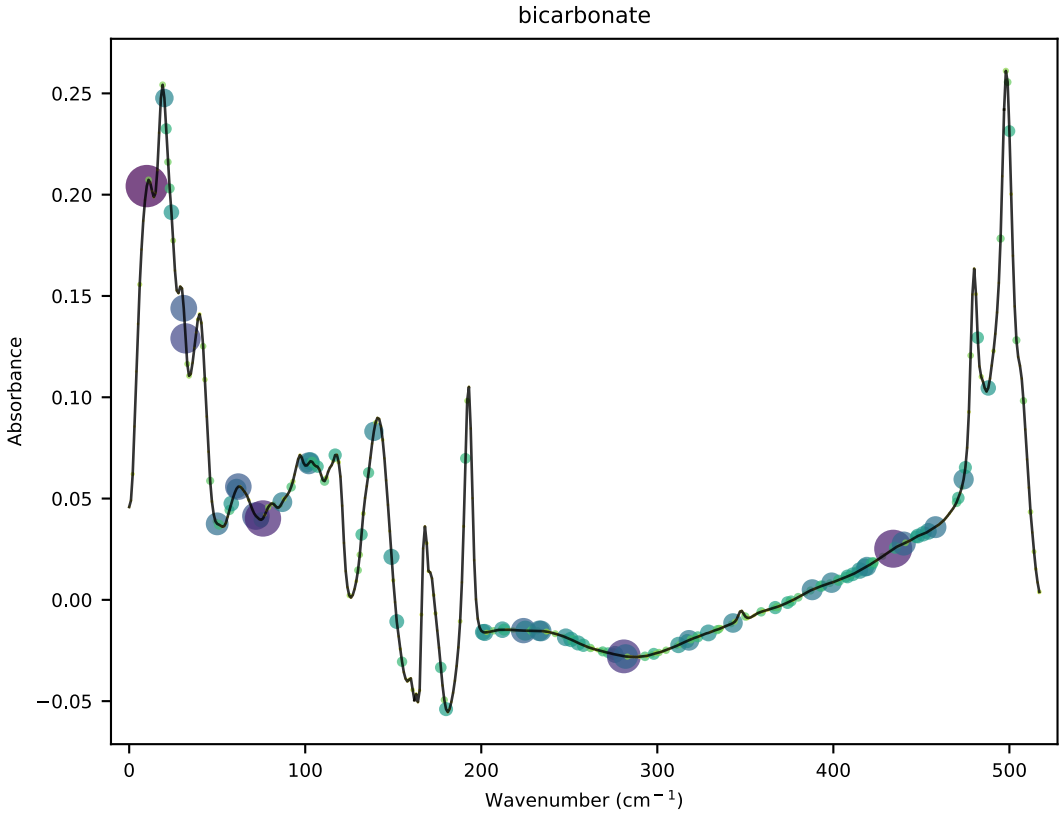

Figura 3.7. Saliências calculadas pela CNN com o espectro de uma amostra adulterada com bicarbonato. As regiões com os destaques maiores e mais escuros representam a intensidade com que cada coordenada contribuiu para a decisão da rede, isto é, as regiões que ativaram o nó final "bicarbonato".

minação automática de regiões espectrais importantes, sem depender de equipamentos e modelos externos. Apesar de os resultados obtidos com a extração de *features* descrita anteriormente não representar ganhos de desempenho dos modelos, os resultados são, em média, equivalentes ao uso das *features* do equipamento FTIR. Além disso, a exploração de saliências é um aspecto importante da interpretabilidade de modelos de aprendizado de máquina.

Tabela 3.4. Valores de acurácia obtidos com a execução do classificador Random Forest (RF) utilizando 60 *features* extraídas automaticamente através da saliência da CNN. As classificações binária e multiclasse foram realizadas nos três conjuntos de treinamento e teste: 90/10%, 75/25% e 50/50%.

Classificação	90/10%	75/25%	50/50%
Multiclasse	0.9072	0.9002	0.8849
Binária	0.9732	0.9579	0.9604

Tabela 3.5. Valores de acurácia obtidos com a execução do classificador Gradient Boosting Machines (GBM) utilizando 60 *features* extraídas automaticamente através da saliência da CNN. As classificações binária e multiclasse foram realizadas nos três conjuntos de treinamento e teste: 90/10%, 75/25% e 50/50%.

Classificação	90/10%	75/25%	50/50%
Multiclasse	0.9031	0.8927	0.8613
Binária	0.9588	0.9554	0.9579

3.6 Resultados e discussão

Ambos os aprendizados, *deep learning* e *ensemble learning* foram avaliados para a detecção de adulterantes nas amostras do leite. O conjunto de dados original contém 4.846 instâncias rotuladas como uma das seis classes possíveis: *leite cru*, *sacarose*, *amido*, *bicarbonato*, *peróxido de hidrogênio* e *formaldeído*. Para a versão multiclasse do problema, todas as seis classes foram consideradas. Já para a versão binária, as classes referentes aos adulterantes foram consideradas como a classe *leite adulterado*, enquanto que a classe *leite cru* foi mantida. As classificações binária e multiclasse foram avaliadas com três pares de conjuntos de treinamento e teste, com diferentes proporções, a fim de avaliar a consistência dos métodos com variações na quantidade de instâncias de treinamento.

Para os métodos *ensemble*, os valores de acurácia nas avaliações variaram de 86,09% a 98,56%. A arquitetura de rede neural convolucional proposta produziu valores de acurácia entre 95,38% a 98,76%. O valor de acurácia média para os modelos RF, GBM e CNN são, respectivamente, 93,23%, 92,25% e 96,76%. Esses resultados mostram que todos os classificadores obtiveram melhor desempenho nas classificações binárias. No entanto, a CNN se mostrou um classificador mais robusto, já que seus resultados com as classificações binária e multiclasse são semelhantes. Todos os resultados de acurácia dos modelos avaliados são apresentados na Tabela 3.6. O detalhamento do resultado de acurácia separado por classe, para o problema multiclasse, é apresentado na Tabela 3.7, onde pode-se notar que todos os classificadores apresentam melhor desempenho na classificação de instâncias na classe *leite cru*, e também que a CNN apresenta melhores resultados de forma geral. É possível ver, também, que o aumento do número de instâncias de treinamento nem sempre melhora o desempenho geral do classificador. Por fim, a CNN é, em geral, mais robusta quando há menos dados para o treinamento do modelo, como, por exemplo, no conjunto de 50% de treinamento.

A área sobre a curva ROC (*Receiver Operating Characteristic*), denominada AUC (*Area Under Curve*) foi avaliada a fim de se obter mais uma medida de desempenho

Tabela 3.6. Valores de acurácia obtidos com a execução dos classificadores Random Forest (RF), Gradient Boosting Machine (GBM) e Rede Neural Convolucional (CNN) para classificações binária e multiclasse. Todos os classificadores foram avaliados com três pares de conjuntos de treinamento e teste, identificados pelas suas proporções de instâncias: 90/10%, 75/25% e 50/50%.

Conjunto de dados	Versão de classificação	RF	GBM	CNN
90/10%	Multiclasse	0.9093	0.8907	0.9608
	Binária	0.9856	0.9711	0.9794
75/25%	Multiclasse	0.8812	0.8787	0.9695
	Binária	0.9744	0.9686	0.9876
50/50%	Multiclasse	0.8700	0.8609	0.9538
	Binária	0.9736	0.9653	0.9546

Tabela 3.7. Valores de acurácia independentemente para cada classe (bicarbonato, formaldeído, peróxido, puro, amido e sacarose) para classificação multiclasse considerando os classificadores Random Forest (RF), Gradient Boosting Machine (GBM) e Rede Neural Convolucional (CNN), para cada conjunto de treinamento e teste: 90/10%, 75/25%, e 50/50%.

Modelo	Conjunto de dados	bicarbonato	formaldeído	peróxido de hidrogênio	leite cru	amido	sacarose
RF	90/10%	0.7804	0.7209	0.7916	0.9883	0.8048	0.9636
	75/25%	0.7540	0.6884	0.7524	0.9839	0.8099	0.8773
	50/50%	0.7634	0.6259	0.7847	0.9805	0.7444	0.8953
GBM	90/10%	0.7804	0.7441	0.7291	0.9766	0.7560	0.9272
	75/25%	0.8032	0.6739	0.7623	0.9759	0.7768	0.8867
	50/50%	0.7551	0.6259	0.7309	0.9780	0.7356	0.8870
CNN	90/10%	0.9756	0.9302	0.8958	0.9844	0.9024	0.9636
	75/25%	0.9918	0.9057	0.9108	0.9887	0.9421	1.0000
	50/50%	0.9958	0.9236	0.8340	0.9861	0.8854	0.9539

dos modelos de classificação. O cálculo foi obtido a partir a partir do *score* AUC de cinco repetições de cada classificador. A Figura 3.8 apresenta as curvas ROC calculadas para cada método na classificação binária e multiclasse. As curvas plotadas indicam um bom desempenho dos modelos em distinguir as classes atribuídas às instâncias, em especial a CNN, que tem o valor de AUC médio próximo de 1 (0.9985 e 0.9971 para o problema binário e multiclasse, respectivamente). Isso significa que o modelo está realizando praticamente todas as predições corretas para cada classe.

Também foram realizados cálculos de *t-test* par-a-par comparando a diferença do AUC médio entre todos os classificadores para os problemas binário e multiclasse. Neste teste, que obtém o valor de *t-value*, quanto maior a magnitude de t, maior a evidência contra a hipótese nula. Isso significa que existe uma maior evidência de

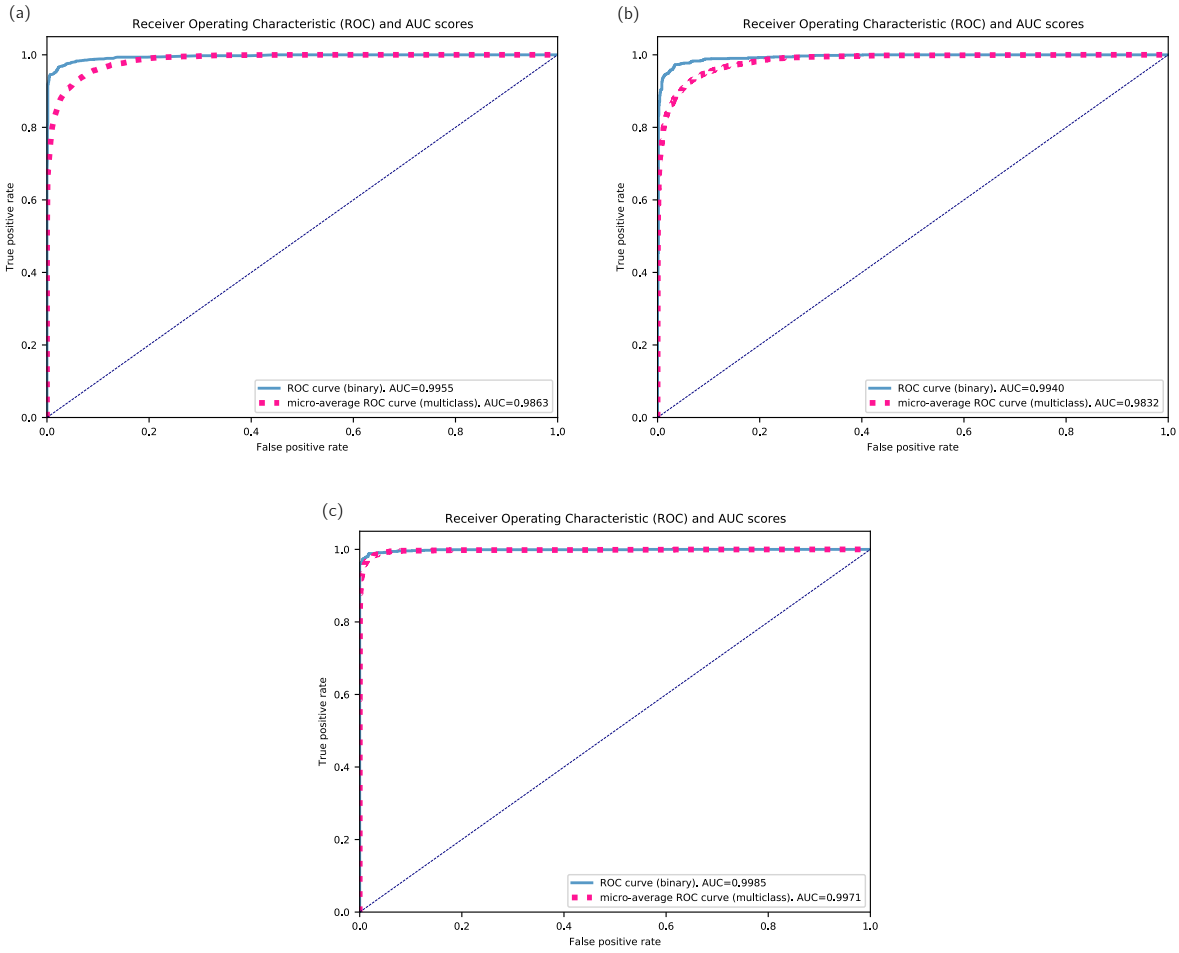

Figura 3.8. Curvas ROC (*Receiver Operating Characteristic*) para as versões binária (linhas contínuas) e multiclasse (linhas tracejadas) dos modelos (a) Random Forest, (b) Gradient Boosting Machine e (c) Rede Neural Convolucional e os *scores* AUC. O valor de ROC multiclasse foi calculado utilizando a estratégia *micro-average*, que soma os valores de verdadeiros positivos, falsos positivos e falsos negativos para todas as classes.

que há uma diferença significativa entre os modelos. Quanto mais próximo t for de 0, maiores as probabilidades de que não há diferença significativa. Quanto maior for o valor absoluto de t, menor será o valor de *p-value*, e maior será a evidência contra a hipótese nula. Os testes de significância estatísticas mostram que a CNN é o classificador mais robusto e possui diferenças significativas de desempenho quando comparados aos métodos *ensemble*. A Figura 3.9 apresenta o *t-test* dos classificadores nas versões binária e multiclasse.

De forma intuitiva, a classificação binária tende a ser um problema mais simples que a classificação multiclasse, pois há menos complexidade na distribuição das instâncias. Isso é observado nos resultados dos modelos de Random Forest e Gradient

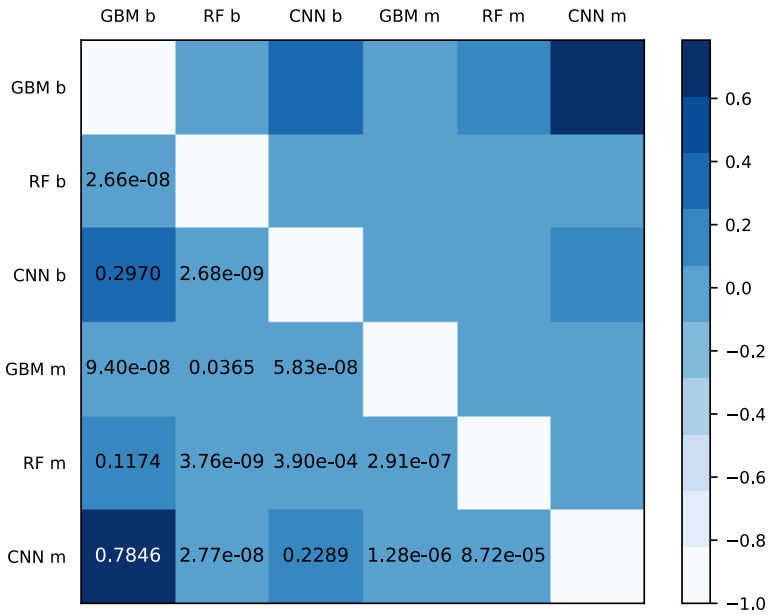

Figura 3.9. Cálculos de *t-test* sobre as diferenças par-a-par do *score* AUC médio para versões binária (b) e multiclasse (m) dos classificadores Random Forest (RF), Gradient Boosting Machine (GBM) e Rede Neural Convolucional (CNN).

Boosting Machines, onde as acurácias de classificação são, no máximo, 10% maiores que as acurácias multiclasse. No entanto, os resultados da Rede Neural Convolucional mostraram que o método é mais robusto na classificação multiclasse, com acurácias um pouco superiores que as das versões binárias. Pode-se concluir, portanto, que as CNN são mais adequadas para a classificação multiclasse para este problema.

É importante destacar que o número de amostras adulteradas no conjunto de dados original é aproximadamente metade do total de amostras. Isso, em termos de classificação binária, resulta em uma distribuição de classes balanceadas (≈50% adulteradas e ≈50% não adulteradas). Por outro lado, quando considera-se a classificação multiclasse, cinco classes diferentes referem-se a amostras adulteradas e a classe *Leite cru* refere-se a amostras puras. A distribuição de ≈50% de amostras puras e ≈10% para cada classe de adulteração leva a uma distribuição de classes desbalanceada. A arquitetura de CNN proposta se mostrou adequada para essa situação, obtendo bons resultados quando há ocorrência de desbalanceamento de classes, como pode ser visto nos valores referentes à classificação multiclasse da Tabela 3.6, com acurácias variando de 95,38% a 98,76%.

Verifica-se, também, que os métodos baseados em aprendizado de máquina, como os *ensemble* de árvores e redes neurais, apresentam desempenhos consideravelmente superiores aos métodos estatísticos tradicionais, geralmente empregados em análises de leite e considerados *benchmark* deste trabalho, como regressão linear, regressão

logística e PLS. O melhor desempenho dos métodos estatísticos tradicionais foi obtido pela regressão logística, que apresentou acurácia de até 76,49% para a classificação binária e 58,76% para a classificação multiclasse.

Os métodos baseados em *deep learning* apresentam desempenhos bastante satisfatórios, especialmente a rede neural convolucional, com acurácia de até 98,76%. Apesar dos bons resultados, uma característica das redes neurais é a dificuldade da interpretação dos modelos, considerando que não há uma relação perceptível dos parâmetros de uma rede, como peso dos nós, número de camadas ou de filtros convolucionais, com a função que está sendo aproximada pela rede [Lipton, 2018]. Além disso, diferentes arquiteturas de redes neurais possuem diferentes habilidades em relação a um mesmo conjunto de dados de entrada. Do ponto de vista de cada instância do conjunto, diferentes modelos podem classificá-la de forma diferente, e uma mesma instância pode ser considerada difícil para um modelo e fácil para outro. Essas características são abordadas no capítulo a seguir, onde se propõe uma metodologia que utiliza uma definição de dificuldade de instância e permite conhecer os limites de capacidade de diferentes modelos, abrindo, eventualmente, a "caixa-preta" dos modelos de redes neurais.

Capítulo 4

Metodologia de aprendizado AutoML

O bom desempenho obtido por uma rede neural convolucional descrito no Capítulo 3 deve-se a uma análise detalhada das características e estruturas das amostras presentes nos conjuntos de dados do leite bovino. A fim de se obter a arquitetura final apresentada naquele capítulo, extensivos testes e adequações manuais foram realizadas e, devido às características de caixa-preta dos modelos *deep learning*, foram necessárias abordagens de tentativa e erro. A arquitetura de rede obtida, portanto, é altamente especializada para resolver problemas relacionados ao leite bovino.

Apesar da precisão dos dados obtidos, compreender o funcionamento detalhado dos modelos gerados ainda é um desafio. Além disso, a generalização da arquitetura para adequação a conjuntos de dados diferentes e problemas de outros domínios é complexa. Recentemente, estudos na área de Aprendizado de Máquina têm focados no desenvolvimento automatizado de arquiteturas de redes neurais para domínios arbitrários. Essas abordagens automatizadas são denominadas Aprendizado de Máquina Automatizado, ou AutoML, e têm obtido resultados interessantes [Hutter et al., 2019]. O bom desempenho desses métodos é, em geral, dependente de técnicas de otimização de hiperparâmetros, de treinamento extensivo de arquiteturas de redes neurais ou mesmo de métodos para aceleração de treinamento dessas. Um sistema AutoML normalmente é projetado a partir de uma análise global dos conjuntos de dados de treinamento disponíveis e, portanto, não consideram características individuais de cada amostra do conjunto de dados, caracterizando apenas métricas relacionadas ao conjunto todo, como a acurácia final de classificação.

Características individuais de instâncias (amostras individuais do conjunto de dados) podem ser aproveitadas por uma metodologia da área de psicometria conhecida como Teoria de Resposta ao Item (*Item Response Theory* - IRT). IRT é um paradigma que estuda a pontuação de testes relacionada ao conjunto de itens de um questionário,

geralmente aplicada para medir as habilidades de um indivíduo, analisando suas respostas sobre um certo domínio. Este raciocínio pode ser estendido para os conceitos de aprendizado de máquina, considerando os modelos como sendo indivíduos e as instâncias do conjunto de dados como sendo os itens do questionário. Informações interessantes podem ser extraídas com o uso de IRT, entre elas a dificuldade e a discriminação de cada instância, que podem caracterizar tanto a complexidade das instâncias quanto as habilidades dos modelos analisados.

Neste capítulo propõe-se o NASirt: uma nova abordagem de aprendizado de máquina baseada em AutoML e na busca por arquitetura neural (NAS), que utiliza informações de complexidade de instâncias de IRT para encontrar arquiteturas de redes neurais de alta precisão para conjuntos de dados espectrais. A metodologia proposta, além de caracterizar informações das instâncias, é capaz de determinar um conjunto de modelos de maior desempenho para o conjunto de dados fornecido, apresentando as configurações de hiperparâmetros e arquiteturas que geram modelos com maiores habilidades.

O capítulo está estruturado como descrito a seguir. Na Seção 4.1, apresentam-se os conceitos do AutoML e suas três divisões principais, otimização de hiperparâmetros, meta-aprendizado e busca por arquitetura neural. Na Seção 4.2 a metodologia NASirt é apresentada, enquanto na Seção 4.3, apresentam-se os experimentos realizados para comparar o desempenho da metodologia em diferentes conjuntos de dados, incluindo detalhes das complexidades do método proposto. Por fim, a Seção 4.4 apresenta uma discussão sobre a metodologia proposta e outros métodos AutoML semelhantes.

4.1 Aprendizado de máquina automatizado (AutoML)

Modelos de aprendizado de máquina estão se tornando cada vez mais poderosos e complexos, atingindo níveis de precisão e desempenho excelentes para as mais variadas áreas de aplicação. Apesar de seu sucesso, modelos de aprendizado de máquina, especialmente modelos de *deep learning*, em geral, são difíceis de treinar e de se compreender. Fazer com que redes neurais, ou qualquer modelo de aprendizado de máquina, aprenda com plena eficiência ainda é um desafio. Além disso, definir a arquitetura desses modelos é um processo feito geralmente de forma manual, através de tentativa e erro e usando o conhecimento de especialistas [Elsken et al., 2019].

Parte da tarefa de projetar uma arquitetura de rede neural está a otimização de hiperparâmetros, que são variáveis que controlam o próprio processo de treinamento, como o número de camadas ocultas utilizadas, bem como quantos nós devem compor

cada camada. A definição de hiperparâmetros adequados, bem como a definição da própria arquitetura, são decisões que devem ser tomadas num processo denominado engenharia de arquitetura ("*architecture engineering*").

O Aprendizado de Máquina Automatizado (*Automated Machine Learning* - AutoML) é uma área do Aprendizado de Máquina que visa tomar decisões da engenharia de arquitetura automática e objetivamente, de forma que uma arquitetura de rede neural e uma abordagem de treinamento adequadas e com alto desempenho sejam propostos pelo sistema de AutoML utilizando como entrada apenas o conjunto de dados fornecido pelo usuário [Hutter et al., 2019]. Assim, o AutoML pode permitir que cientistas utilizem modelos de aprendizado de máquina extremamente precisos, especialmente com *deep learning*, sem a necessidade de entender em detalhes a tecnologia por trás do seu funcionamento.

O AutoML pode ser divido em três grandes tarefas: otimização de hiperparâmetros, meta-aprendizado e busca por arquitetura neural (*Neural Architecture Search* - NAS). Essas tarefas são descritas a seguir.

4.1.1 Otimização de hiperparâmetros

A tarefa mais básica de um método AutoML é definir os hiperparâmetros de forma a otimizar o desempenho do modelo. A otimização de hiperparâmetros visa reduzir o esforço humano para se aplicar o aprendizado de máquina através da busca automatizada de conjuntos de hiperparâmetros. Em geral, essa busca pode ser descrita como um dos métodos: busca em grade (*grid search*) e busca aleatória (*random search*). Ambos os métodos de busca visam encontrar mínimos globais na função de perda (*loss*) do modelo de aprendizado de máquina [Bergstra & Bengio, 2012].

Grid search é o método mais básico para a otimização de hiperparâmetros, onde o usuário especifica um conjunto finito de valores possíveis para cada hiperparâmetro e a busca é feita através do produto cartesiano desses valores [Hutter et al., 2019]. Este método, apesar de simples e bastante utilizado, sofre com o problema de dimensionalidade, já que o número de funções avaliadas cresce exponencialmente de acordo com a dimensão do espaço de busca (número de configurações).

Uma alternativa comum ao *grid search* é o método *random search*, que busca por configurações de hiperparâmetros aleatoriamente, até que algum limite para a busca seja atingido. A busca aleatória funciona melhor quando alguns dos hiperparâmetros são mais relevantes que outros, uma propriedade frequente em modelos de aprendizado de máquina [Hutter et al., 2019].

Em comparação com a *grid search*, a busca aleatória pode avaliar mais parâmetros

que a busca em grade, uma vez que não há a necessidade de combinação de todas as possibilidades. Assim, o espaço a ser explorado na busca por mínimos da função pode ser mais amplo. A Figura 4.1 ilustra essa diferença. No entanto, a *grid search* oferece maior controle das possíveis configurações a serem testadas [Hutter et al., 2019].

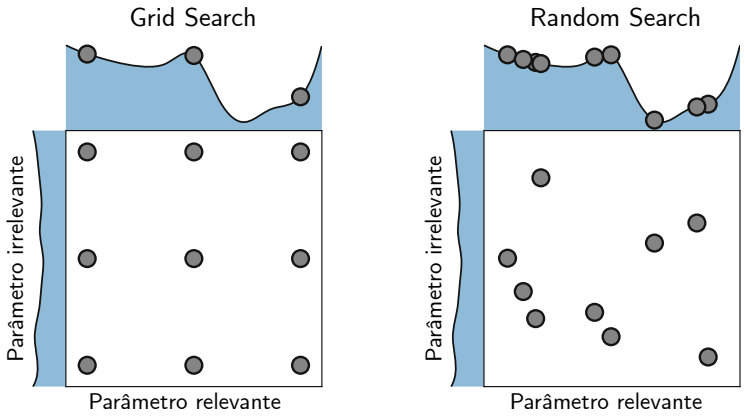

Figura 4.1. Comparação de *grid search* e *random search* para a minimização de uma função. Figura baseada nas ilustrações de Hutter et al. [2019] e Bergstra & Bengio [2012].

4.1.2 Meta-aprendizado

O meta-aprendizado é considerado uma tarefa de AutoML, uma vez que visa o desenvolvimento de métodos que aprendem através da experiência em outras tarefas de aprendizado. O termo meta-aprendizado envolve qualquer tipo de aprendizado baseado em experiências prévias com outras tarefas, como a avaliação de modelos de aprendizado de máquina independentes.

Primeiramente, um modelo de meta-aprendizado deve coletar meta-dados que descrevem tarefas de aprendizado anteriores, bem como os modelos aprendidos. Esses dados compreendem as configurações exatas utilizadas para treinar os modelos, incluindo os hiperparâmetros utilizados, composições de *pipelines* e arquiteturas de rede. Além disso, devem ser levados em consideração os resultados de avaliações dos modelos, como acurácias obtidas, tempos de treinamento e os parâmetros aprendidos (pesos da rede). Em seguida, deve-se aprender a partir desses meta-dados, extrair e transferir o conhecimento que guia a busca por melhores modelos considerando novas tarefas [Hutter et al., 2019]. Um exemplo de modelo de meta-aprendizado pra um problema de classificação seria a avaliação de modelos individuais como um classificador de árvore de decisão, uma regressão logística e uma CNN. Posteriormente, uma outra rede neural é treinada como um meta-modelo e a rede tomará como entrada as saídas

dos três modelos individuais. A rede aprenderá a retornar as predições finais como uma função dos modelos.

O meta-aprendizado, portanto, utiliza a experiência da avaliação de modelos de aprendizado de máquina para aprender a realizar tarefas com dados novos. As técnicas abordadas se baseiam nos processos que ocorrem quando o ser humano aprende novas habilidades, isto é, raramente aprende-se tudo do zero. Inicia-se, sempre, com habilidades aprendidas anteriormente em tarefas relacionadas, reutilizando-se abordagens que funcionaram bem antes. Abordagens de meta-aprendizado podem diminuir bruscamente o custo necessário para se obter bom desempenho em tarefas de aprendizado de máquina completamente novas [Hutter et al., 2019].

4.1.3 Neural Architecture Search

A busca por arquitetura neural, ou *Neural Architecture Search* (NAS), é o processo de automatização da engenharia de arquitetura (*architecture engineering*) de redes neurais artificiais. Os conceitos de NAS possuem se mesclam significativamente com os conceitos da otimização de hiperparâmetros e do meta-aprendizado. Um procedimento baseado em NAS pode ser dividido em três componentes principais: espaço de busca, estratégia de busca e estratégia de estimativa de desempenho [Elsken et al., 2019; Kyriakides & Margaritis, 2020].

Em uma abordagem NAS, o espaço de busca define as redes que podem ser examinadas para produzir uma arquitetura final. A definição de um espaço de busca de qualidade permite a obtenção de arquiteturas com alto desempenho, mas é necessário conhecimentos prévios sobre o conjunto de dados analisado. O espaço de busca de arquiteturas pode ser impraticável e pode-se limitar o seu tamanho ao incorporar conhecimentos prévios sobre propriedades adequadas para uma tarefa. Isso simplifica a busca, mas também introduz um viés humano, pois exclui arquiteturas não exploradas, que poderiam oferecer melhores desempenhos [Elsken et al., 2019].

A estratégia de busca faz um detalhamento de como explorar o espaço de busca, o que pode influenciar fortemente a eficiência da busca e a efetividade geral da arquitetura proposta. Escolher uma estratégia de busca apropriada pode garantir que o espaço de busca seja suficientemente explorado. Muitas estratégias diferentes podem ser usadas para explorar o espaço de arquiteturas, como a busca aleatória (*random search*), otimização Bayesiana, métodos evolucionários, aprendizado por reforço, entre outros. A otimização Bayesiana, por exemplo, vem sendo utilizada desde 2013 para arquiteturas orientadas à visão computacional, obtendo as primeiras redes neurais geradas automaticamente a vencer competições contra humanos em conjuntos de dados

conhecidos, como o CIFAR-10 [Hutter et al., 2019].

Por fim, a estratégia de estimativa de desempenho é responsável por comparar resultados intermediários e ajudar a estratégia de otimização a escolher uma dentre várias opções durante a fase de busca. Na comparação de resultados, deve-se obter estimativas de desempenho, e a maneira mais simples de se obter esses resultados é através do treinamento de validação da arquitetura com os dados disponíveis. No entanto, avaliar essas propriedades de modelos *deep learning* é um processo computacionalmente caro, o que limita o número de arquiteturas que podem ser exploradas [Elsken et al., 2019; Kyriakides & Margaritis, 2020]. Estudos mais recentes na área focam em desenvolver métodos que reduzem o custo dessas estimativas de desempenho, como a utilização de estimativas baseadas em baixa fidelidade da avaliação real (treinamentos com subconjuntos dos dados, imagens em resolução mais baixa, menos filtros por cama, etc). Outra maneira de acelerar a estimativa de desempenho é inicializar os pesos das arquiteturas com base nos pesos de arquiteturas que já tenham sido treinadas antes. Este processo é denominado morfismo de rede (*network morphism*) e permite modificar uma arquitetura visando manter intacta a função original da rede [Wei et al., 2016; Jin et al., 2019].

4.2 Metodologia NASirt

Neste trabalho, propõe-se uma metodologia de AutoML que utiliza uma abordagem de busca por arquitetura neural (NAS). A metodologia denomina-se NASirt, e emprega a Teoria de Resposta ao Item com modelos de Redes Neurais Convolucionais para oferecer um melhor entendimento de amostras, especificamente daquelas provenientes de análises espectrais. A proposta da metodologia é oferecer uma maior compreensão das habilidades de aprendizado desses modelos de Aprendizado de Máquina. O NASirt é focado em problemas de classificação e seu objetivo é encontrar arquiteturas e combinações de hiperparâmetros de Redes Neurais Convolucionais (CNN) que maximizam o desempenho de predição em conjuntos de dados espectrais. Além disso, o método permite obter uma visão da dificuldade das instâncias analisadas, demonstrando quais instâncias que influenciam na qualidade das predições dos modelos.

4.2.1 Teoria de Resposta ao Item e Aprendizado de Máquina

Os conceitos teóricos gerais da Teoria de Resposta ao Item foram apresentados na Seção 2.5, Capítulo 2, página 26. As ferramentas apresentadas pela teoria permitem analisar habilidades de indivíduos em relação às suas respostas para itens de um ques-

tionário, bem como caracterizar a complexidade desses itens. Apesar de concebidos para indivíduos e questionários, os conceitos da IRT podem ser generalizados para outros domínios.

A Teoria de Resposta ao Item pode ser abordada no contexto de Aprendizado de Máquina, de forma a associar as entidades item, indivíduo examinado e questionário às noções de instância, modelo de aprendizado de máquina e conjunto de dados, respectivamente. Em outras palavras, os itens de um questionário podem ser considerados instâncias pertencentes a um conjunto de dados, enquanto o indivíduo que responde ao questionário pode ser considerado o modelo inferido a partir de um treinamento. De fato, estudos recentes têm associado IRT ao Aprendizado de Máquina e têm obtido resultados interessantes [Martínez-Plumed et al., 2019; Cardoso et al., 2020; Lalor et al., 2016, 2018]. O estudo de Martínez-Plumed et al. [2019] apresenta, de forma aprofundada, a união entre IRT e Aprendizado de Máquina, e é utilizado como uma importante referência deste trabalho. Porém, os autores comparam o uso de IRT em diferentes classificadores e apresentam uma abordagem teórica que analisa os parâmetros de IRT e as habilidades dos modelos, enquanto na metodologia proposta neste trabalho, modelos de IRT são utilizados para apoiar a geração automatizada de modelos de Aprendizado de Máquina, a fim de produzir um novo método de classificação.

Os parâmetros dos modelos de IRT também podem ter suas noções associadas ao Aprendizado de Máquina. As dificuldades associadas aos itens podem ser entendidas como a complexidade de uma amostra: caso o modelo apresente um certo grau de confiança na predição de uma instância, essa instância pode ser considerada difícil. Além disso, uma mesma instância pode levar a predições diferentes para modelos diferentes, característica que pode ser entendida como a discriminação em IRT. Assim, da mesma forma que a IRT oferece uma visão detalhada das capacidades inerentes dos indivíduos examinados, os modelos de aprendizado de máquina podem se beneficiar destas análises. A abordagem de IRT pode descrever habilidades especialmente de modelos de redes neurais artificiais, considerados complexos e de difícil compreensão.

4.2.2 Etapas da metodologia

De modo geral, o NASirt depende de treinamentos independentes de modelos CNN baseados no método de validação *hold-out*, descrito na Seção 2.4.2. Inicialmente, como primeiro passo da metodologia, define-se a divisão do conjunto de dados em "*folds*", como sugere a validação *hold-out*. A fim de se demonstrar consistência nos resultados, diferentes *folds* são utilizadas, como por exemplo, 90% para treinamento e 10% para teste, 75% para treinamento e 25% para teste e 50% para treinamento e 50% para

teste. As instâncias pertencentes a cada *fold* são selecionadas aleatoriamente a partir do conjunto de dados original. As etapas da metodologia descritas a seguir são repetidas independentemente para uma *fold* de treinamento e teste específica.

As etapas descritas são referentes a uma *fold* de treinamento e teste específica. Inicialmente, realiza-se o treinamento de uma coleção de modelos de CNN com variações de arquiteturas e hiperparâmetros. Nesta etapa, é definido um conjunto de hiperparâmetros que serão explorados de forma a testar todas as combinações de hiperparâmetros, gerando m modelos distintos. As variações exploradas incluem o número de filtros convolucionais, tamanho de *kernel*, taxa de *dropout*, tamanho da camada totalmente conectada etc. Então, submete-se as instâncias do conjunto de teste para todos os m modelos treinados, armazenando-se a predição de cada modelo para cada instância. As predições armazenadas de todos os modelos são utilizadas para gerar o modelo de IRT, uma vez que a estrutura é semelhante às "respostas" de um questionário. O modelo de IRT oferece importantes parâmetros sobre as instâncias do conjunto, como dificuldade e discriminação. O valor de *true score* também é calculado. Em seguida, os valores obtidos pela IRT são normalizados para valores entre 0 e 1.

De acordo com o trabalho de Martínez-Plumed et al. [2019], a fim de evitar a ocorrência de singularidade no modelo IRT, isto é, uma possibilidade de que todos os classificadores acertem a predição para todas as instâncias, é recomendado introduzir classificadores artificiais específicos. Então, além dos modelos treinados, deve-se introduzir classificadores artificiais para que o modelo de IRT não sofra com singularidade nas respostas [Martínez-Plumed et al., 2019]: três classificadores aleatórios, cujas predições são obtidas ao acaso, desconsiderando quaisquer informações presentes nas amostras, um classificador otimista e um classificador pessimista que, respectivamente, sempre acertam e sempre erram todas as predições.

O valor de *true score*, descrito anteriormente na Seção 2.5.2, Capítulo 2, página 28, neste caso, é utilizado para elencar a coleção de modelos treinados de acordo com suas habilidades considerando o conjunto de dados de teste. Martínez-Plumed et al. [2019] afirmam que modelos com altos valores de *true score* normalmente têm bom desempenho de acurácia na execução dos testes de instâncias. Portanto, o próximo passo da metodologia proposta é criar um *ranking* compondo um número de n modelos com maiores *true scores* da coleção. Em outras palavras, esse *ranking* representa os n modelos mais habilidosos especificamente para as instâncias de teste, então, é seguro descartar todos os modelos restantes treinados na coleção inicial. Sugere-se, para execução adequada da metodologia, a escolha de um valor de n relativamente pequeno, como $n = 5$ ou $n = 10$ modelos.

A partir de agora, em vez de testar todos os modelos treinados com as instâncias

do conjunto de teste, pode-se utilizar as informações de dificuldade e discriminação para separar as instâncias em grupos e executá-las com modelos específicos do *ranking* criado. É mais provável que o modelo com a maior habilidade seja mais apto a predizer, com maior desempenho, instâncias mais difíceis do conjunto, enquanto que instâncias mais fáceis podem ser submetidas a modelos com menores habilidades. A mesma ideia pode ser aplicada para a discriminação.

Seleciona-se, portanto, um dos dois parâmetros de IRT durante a execução da metodologia: dificuldade (b) ou discriminação (a). Realiza-se uma ordenação de instâncias conforme o parâmetro selecionado, isto é, a lista de instâncias é ordenada da menor dificuldade para a maior, se o parâmetro b for selecionado, ou da menor discriminação para a maior, se o parâmetro a for selecionado. Divide-se essas instâncias ordenadas em n grupos, a mesma quantidade de modelos definida anteriormente, de forma que cada grupo tenha aproximadamente o mesmo número de instâncias. Como as instâncias estão ordenadas, o primeiro grupo possui as instâncias mais fáceis, se o parâmetro escolhido for dificuldade, ou menos discriminativas, se o parâmetro escolhido for discriminação. Cada grupo na sequência possui instâncias cada vez mais difíceis (ou mais discriminativas). Então, realiza-se o teste de classificação com cada um dos n modelos especificamente com um dos n grupos de instâncias, de forma que o primeiro modelo, com *true score* mais baixo, classifique as instâncias do grupo menos difícil (ou menos discriminativo), o segundo modelo classifique as instâncias do segundo grupo, e assim por diante.

Finalmente, as predições realizadas pelos modelos em cada grupo de instâncias são consideradas para calcular a acurácia individual de cada modelo. O último passo é a consolidação da acurácia final, considerando os valores preditos por cada um dos n modelos. Nesta etapa de consolidação, que calcula a acurácia final para o método como um todo, soma-se o número de predições corretas em cada grupo de instâncias, obtendo-se o número total de acertos, e divide-se este número pela quantidade total de instâncias existentes no conjunto de dados. Essa consolidação é a predição final do método proposto, para um *fold* de treinamento e teste específico. A Figura 4.2 apresenta um diagrama dos passos principais descritos como parte do método proposto.

Os passos listados a seguir resumem brevemente a metodologia proposta.

Passo 1. Criar uma coleção de modelos CNN com variações de arquiteturas e hiperparâmetros predefinida, obtendo um total de m modelos;

Passo 2. Submeter todas as instâncias do conjunto de dados de teste a todos os m modelos da coleção;

Passo 3. Considerar as predições das instâncias de cada modelo como respostas de um questionário para a geração do modelo de IRT;

Passo 4. Normalizar os parâmetros de IRT obtidos (dificuldade e discriminação);

Passo 5. Selecionar os n melhores modelos baseando-se nos valores de *true score* obtidos de IRT;

Passo 6. Definir um parâmetro de IRT a ser utilizado (dificuldade ou discriminação) e dividir as instâncias de teste em n grupos de instâncias;

Passo 7. Executar a classificação de cada um dos n modelos considerando um grupo de instâncias específico, de forma que modelos mais habilidosos classifiquem instâncias mais difíceis (ou mais discriminativas);

Passo 8. Calcular a acurácia de cada modelo;

Passo 9. Consolidar os valores de acurácia.

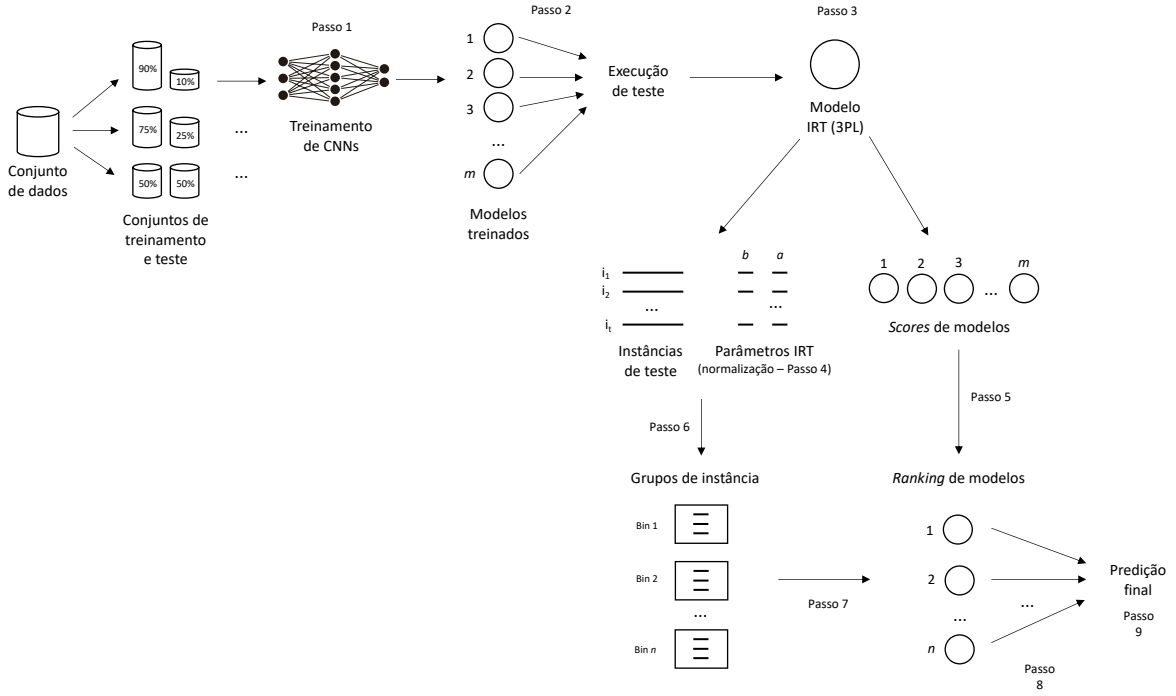

Figura 4.2. Visão geral das etapas da metodologia de AutoML proposta, denominada NASirt.

4.3 Experimentos

Com o intuito de demonstrar as características da metodologia proposta, foram conduzidos diversos experimentos utilizando diferentes conjuntos de dados espectrais, considerando variações do método de avaliação *hold-out*. Cada experimento é relacionado com um conjunto de dados específico e inclui resultados de comparações com métodos utilizados como *benchmark*. Durante a condução dos experimentos, o método NA-

Sirt proposto é comparado com um modelo único de CNN de alto desempenho, com uma abordagem de votação da maioria (*majority-voting*) que combina os resultados de classificação de todos os modelos treinados e, por fim, é comparado também com um execuções da ferramenta de AutoML denominada Auto-Keras.

O objetivo dos experimentos conduzidos é demonstrar que a metodologia NASirt proposta é capaz de atingir desempenho pelo menos semelhante aos métodos apresentados como *benchmark*. Ao contrário do modelo único de CNN, cuja arquitetura é desenvolvida especificamente para um problema, e da votação, que considera predições de todos os modelos, o método apresentado seleciona uma combinação adequada de combinação de hiperparâmetros de uma forma automatizada. Além disso, a metodologia proposta traz informações adicionais de qualidade de instância e habilidade de modelos, fornecidas pela abordagem de IRT. Por isso, mesmo a obtenção de uma similaridade dos resultados indica que a metodologia NASirt é a mais adequada.

Além das capacidades de predição de alto desempenho, o método de AutoML proposto produz dados de apoio interessantes, relacionados com a explicabilidade dos modelos e as complexidades de instâncias. Essa informação pode ser utilizada para obter um melhor entendimento dos conjuntos de dados analisados e características específicas de suas instâncias. Em outras palavras, baseando-se nos dados produzidos pela metodologia, pode-se caracterizar o conjunto de dados no nível de instâncias, podendo comparar diretamente instâncias ao inferir quantitativamente suas dificuldades e suas capacidades discriminativas. Assim, para cada conjunto de dados, são fornecidas informações adicionais, além da acurácia de classificação final, que são apresentadas e descritas na condução de cada experimento.

O treinamento da coleção de modelos CNN inicial considera a mesma variação de hiperparâmetros em cada experimento. Os hiperparâmetros considerados para o treinamento das redes são o número de camadas e de filtros convolucionais, o tamanho de *kernel* (comprimento da janela de convolução), o número de camadas densas (totalmente conectadas), o número de neurônios m cada camada densa, a taxa de *dropout* [Srivastava et al., 2014], o tamanho da janela de *max-pooling* e o tipo de ativação das camadas [Maas et al., 2013]. Os modelos de CNN foram treinadas com todas as combinações possíveis de valores desses hiperparâmetro, listadas na Tabela 4.1. Algumas características foram mantidas fixas a fim de gerar as arquiteturas, como a ocorrência de normalizações de camadas [Ioffe & Szegedy, 2015], o valor de coeficiente de inclinação foi mantido em 0.3 no caso de escolhida a camada de ativação LeakyReLU e o otimizador Adam [Kingma & Ba, 2014] foi utilizado em todas as combinações.

Os experimentos foram conduzidos de forma que, a cada conjunto de dados, considerando o método de validação *hold-out*, três proporções (*folds*) de subconjuntos

de treinamento e teste foram definidos: 90% para treinamento e 10% para teste, 75% para treinamento e 25% para teste e 50% para treinamento e 50% para teste. Para cada conjunto de dados e cada *fold*, 384 modelos diferentes foram gerados através da combinação de todos os hiperparâmetros definidos, conforme o Passo 1 da metodologia proposta. As arquiteturas de CNN foram implementadas utilizando Keras [Chollet et al., 2015] e TensorFlow [Abadi et al., 2016], em Python. As análises baseadas em IRT foram obtidas baseando-se no pacote MIRT em R [Chalmers, 2012] e a integração com os códigos em Python foi realizada utilizando a biblioteca 'rpy2'.

Tabela 4.1. Valores de hiperparâmetros utilizados para gerar a coleção de CNN como primeiro passo da metodologia.

Hiperparâmetro	Descrição	Valores possíveis
Camadas convolucionais	Número de camadas convolucionais	1, 2
Filtros convolucionais	Número de filtros convolucionais que aprendem características da entrada	8, 32, 128
Tamanho de *kernel*	Tamanho de *kernel* convolucional, isto é, o comprimento da janela de convolução	8, 16
Camadas densas	Número de camadas totalmente conectadas inseridas após as camadas convolucionais	1, 2
Tamanho de camada densa	Número de neurônios que compõem cada camada totalmente conectada	128, 1024
Taxa de *dropout*	Taxa de neurônios ignorados aleatoriamente a fim de prevenir *overfitting* na rede	0, 0.4
Tamanho de *max-pooling*	Tamanho da janela de *max-pooling*, que reduz a resolução da entrada ao obter o valor máximo considerando a extensão da janela	0, 4
Ativação	A função de ativação realizada, isto é, a função que define a saída de um neurônio dada uma entrada em uma camada	`leakyrelu`, `tanh`

4.3.1 Seleção de conjuntos de dados

Para a realização dos experimentos, foram coletados três conjuntos de dados diferentes contendo amostras espectrais obtidas de diferentes fontes. O conjunto de dados 1 é denominado "Adulterantes do Leite" e contém espectros infravermelhos de milhares de

amostras de leite bovino puras ou adulteradas. O conjunto de dados 2 é denominado "Soro do Leite" e também contém espectros de leite, mas foram acrescentadas apenas o soro da produção do queijo em diferentes proporções. O conjunto de dados 3 é denominado "Árvores" e contém leituras espectrais de diferentes tecidos de espécies de árvores variadas. A Tabela 4.2 apresenta o número de instâncias, *features* e classes para cada conjunto de dados, bem como as distribuições das classes. Como os conjuntos de dados referem-se a informações espectrais, o número de *features* refere-se ao número de pontos, ou de coordenadas, que formam cada espectro. Este número é relacionado com a precisão, ou resolução, no qual o método de espectroscopia foi aplicado nas amostras para a obtenção dos dados.

Tabela 4.2. Conjuntos de dados utilizados para os experimentos da metodologia NASirt proposta.

ID	Conjunto de dados	# Instâncias	# *Features*	# Classes	Distribuição de classes
1	Adulterantes do Leite	4846	518	6	(50.1%, 9.5%, 10.0%, 9.6%, 9.9%, 10%)
2	Soro do Leite	1040	518	2	(50%, 50%)
3	Árvores	1270	1154	4	(20%, 24.4%, 28.1%, 27.5%)

Os conjuntos de dados 1 e 2 foram criados a partir das amostras fornecidas pelo Laboratório de Análise da Qualidade do Leite da Escola de Veterinária da Universidade Federal de Minas Gerais (EV-UFMG). O conjunto de dados 1 compreende as amostras apresentadas anteriormente na Seção 3.3, Capítulo 3, página 35, em que se descreve a adulteração das amostras com uma de cinco substâncias conhecidas (bicarbonato, formol, peróxido, amido e sacarose), considerando amostras puras ou adulteradas. O experimento que utilizou este conjunto de dados para avaliar a metodologia NASirt considerou a classificação multiclasse para determinar a substância presente nas amostras, quando adulteradas, ou a ausência de adulterantes. A abordagem de IRT permite a determinação dos parâmetros dificuldade e discriminação das instâncias desse conjunto, que são apresentadas na Figura 4.3, para cada *fold* de treinamento e teste analisado. Na proporção de 50/50%, pode-se visualizar uma distribuição mais uniforme para os parâmetros de IRT.

O conjunto de dados 2 foi obtido de maneira semelhante ao conjunto de dados 1, mas ele consiste de amostras diferentes, que foram adulteradas com soro de leite bovino. O soro é considerado um tipo diferente de adulterante, já que é um subproduto obtido a partir da produção do queijo a partir do próprio leite. Em aproximadamente metade das amostras foram adicionadas quantidades variadas de soro de leite, enquanto a outra metade foi mantida como leite puro. Neste experimento, foi realizada a classificação binária para determinar se as amostras são referentes ao leite puro ou ao leite adulte-

rado. Na Figura 4.4, para cada *fold* de treinamento e teste, apresentam-se a dificuldade e a discriminação de IRT das amostras desse conjunto e, partir da distribuição desses valores, pode-se verificar que as dificuldades das instâncias neste caso são semelhantes ao conjunto de dados 1, mas as amostras mais discriminativas são distribuídas mais uniformemente.

Já o conjunto de dados 3 foi produzido a partir de amostras disponíveis publicamente [Ramirez et al., 2015] e contém amostras de espectro infravermelho de órgãos de árvores (raiz, tronco, galhos e folhas), obtidas a partir de 73 espécies de árvores em biomas tropicais e temperados, pertencentes a vários tipos de florestas. Neste experimento foi realizada a classificação multiclasse considerando o órgão de origem das amostras. Os parâmetros de IRT, apresentados na Figura 4.5, mostram que há, especialmente nos *folds* 75/25% e 50/50%, relativamente mais instâncias difíceis do que nos conjuntos anteriores.

Além das informações específicas de dificuldade e discriminação das amostras, a utilização da IRT na metodologia permite, ainda, apresentar visualmente as Curvas de Característica do Item (ICC), que descrevem a relação entre os níveis de habilidades de modelos (θ) e as probabilidades de resposta correta ao item ($P(\theta)$), conforme descrito anteriormente na Seção 2.5.1, Capítulo 2, página 27. A Figura 4.6 apresenta essas curvas, representando todas as classes, em cada um dos conjuntos de dados disponíveis.

Na Figura 4.6-(A), referente ao conjunto de dados Adulterantes do Leite, pode-se notar que as instâncias representadas pelas curvas de formol e bicarbonato têm mais probabilidade de acerto com modelos de maior habilidade, enquanto as instâncias restantes têm mais chance de acerto com modelos menos habilidosos. Além disso, as curvas de formo e bicarbonato são mais íngremes, significando maiores valores de discriminação em relação aos itens restantes. A Figura 4.6-(B) descreve duas curvas de cada classe: puro e soro, que representam itens relativamente fáceis, com alta probabilidade de acerto, mesmo para modelos pouco habilidosos. Neste gráfico, duas curvas do tipo "puro" apresentam diferentes características discriminativas, uma vez que suas inclinações são diferentes. A curva "puro" mais à esquerda representa uma instância menos discriminativa. Por fim, na Figura 4.6-(C), o item da classe "galho" é mais fácil que os demais, porém é menos discriminativo. As curvas restantes mais à direita são mais íngremes e, portanto, mais discriminativas.

Como sugerido pelo trabalho de Martínez-Plumed et al. [2019], apresenta-se o relacionamento entre a habilidade estimada pelo modelo de IRT e a acurácia de classificação para cada conjunto de dados experimentado na Figura 4.7. A figura mostra esse relacionamento para cada conjunto de dados e para cada *fold* de treinamento e teste. As linhas mostram o relacionamento de cada conjunto: (A) Adulterantes do

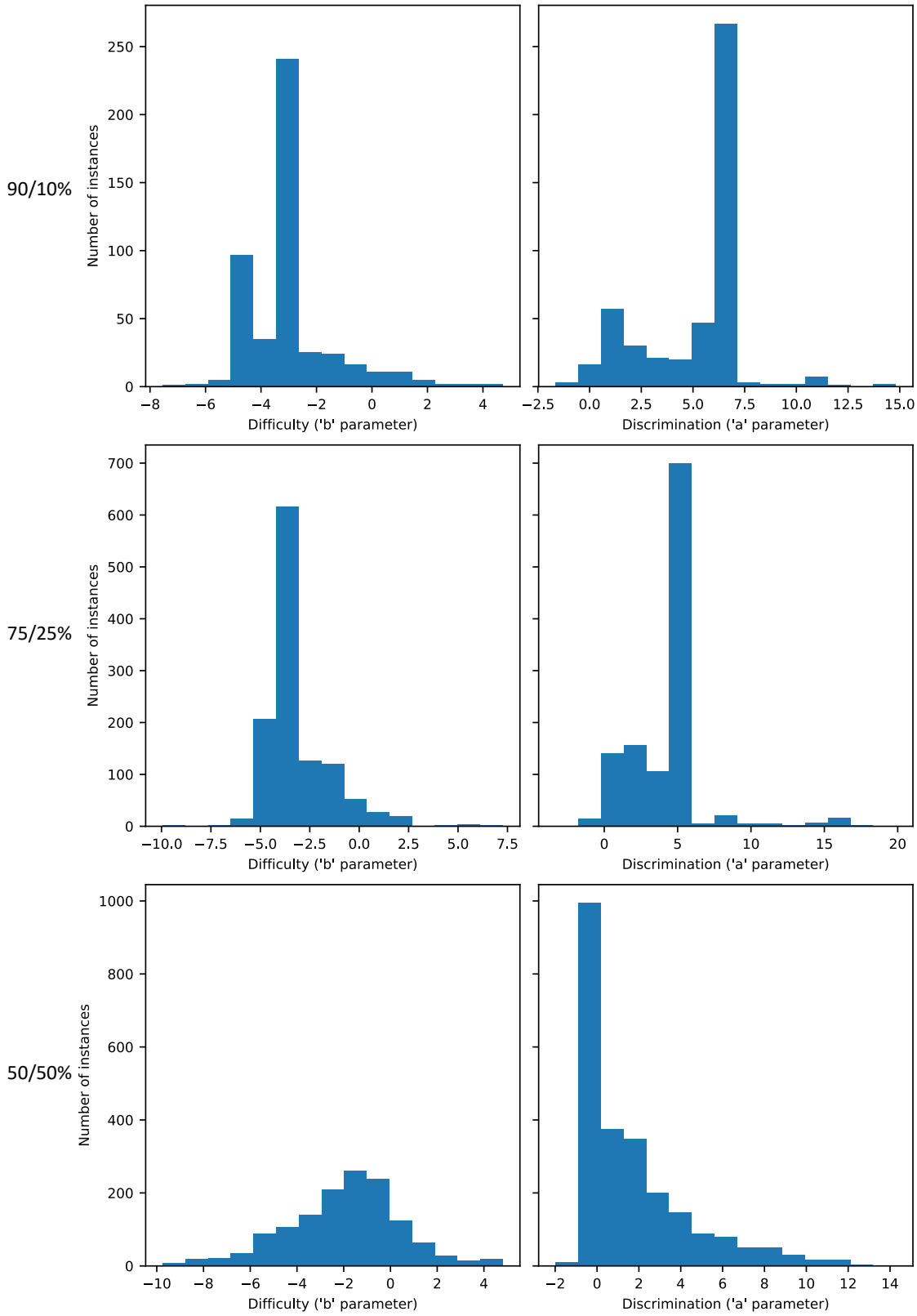

Figura 4.3. Visualização de dificuldade e discriminação de instâncias do conjunto de dados Adulterantes do Leite, para todas as proporções de treinamento e teste.

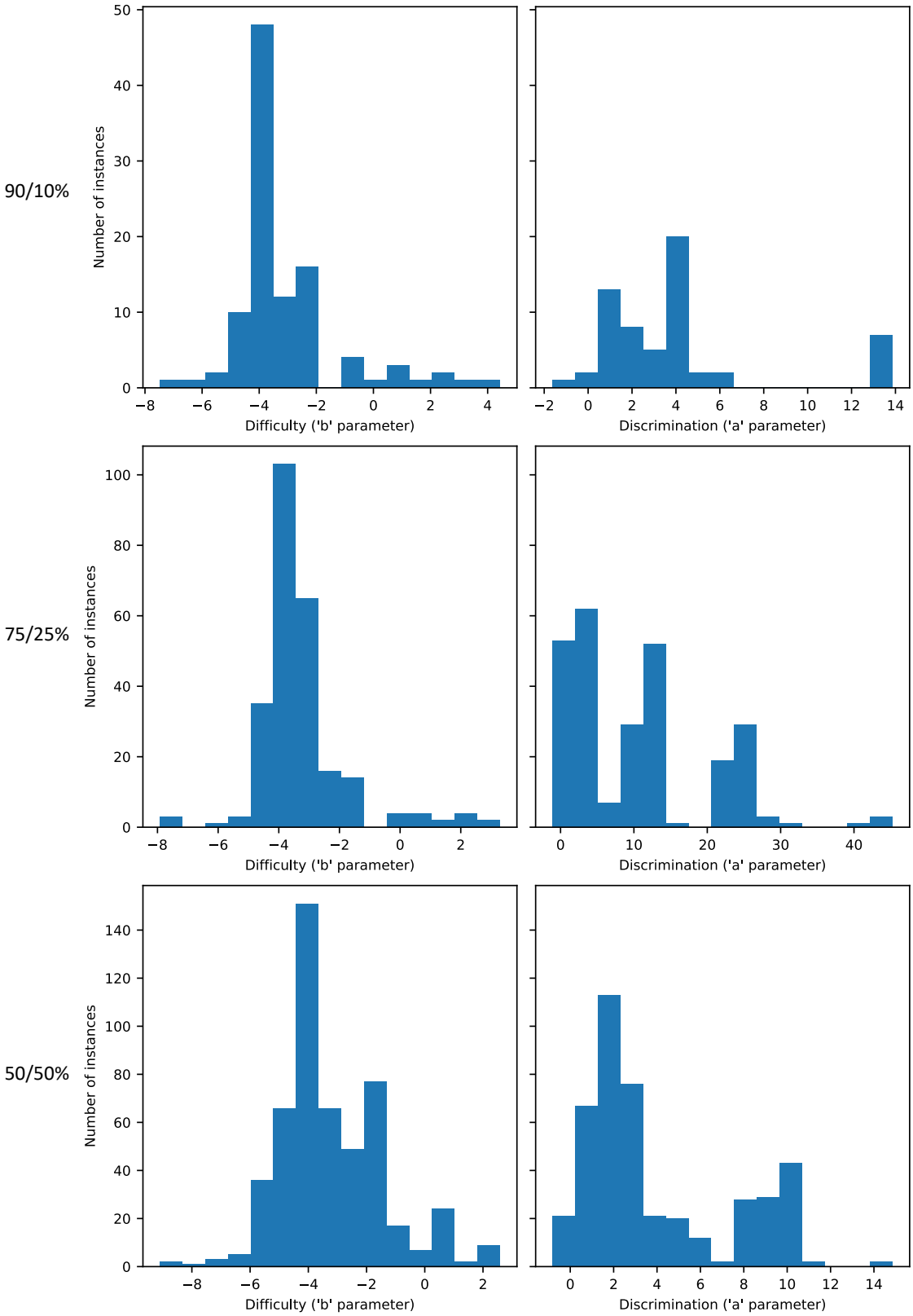

Figura 4.4. Visualização de dificuldade e discriminação de instâncias do conjunto de dados Soro do Leite, para todas as proporções de treinamento e teste.

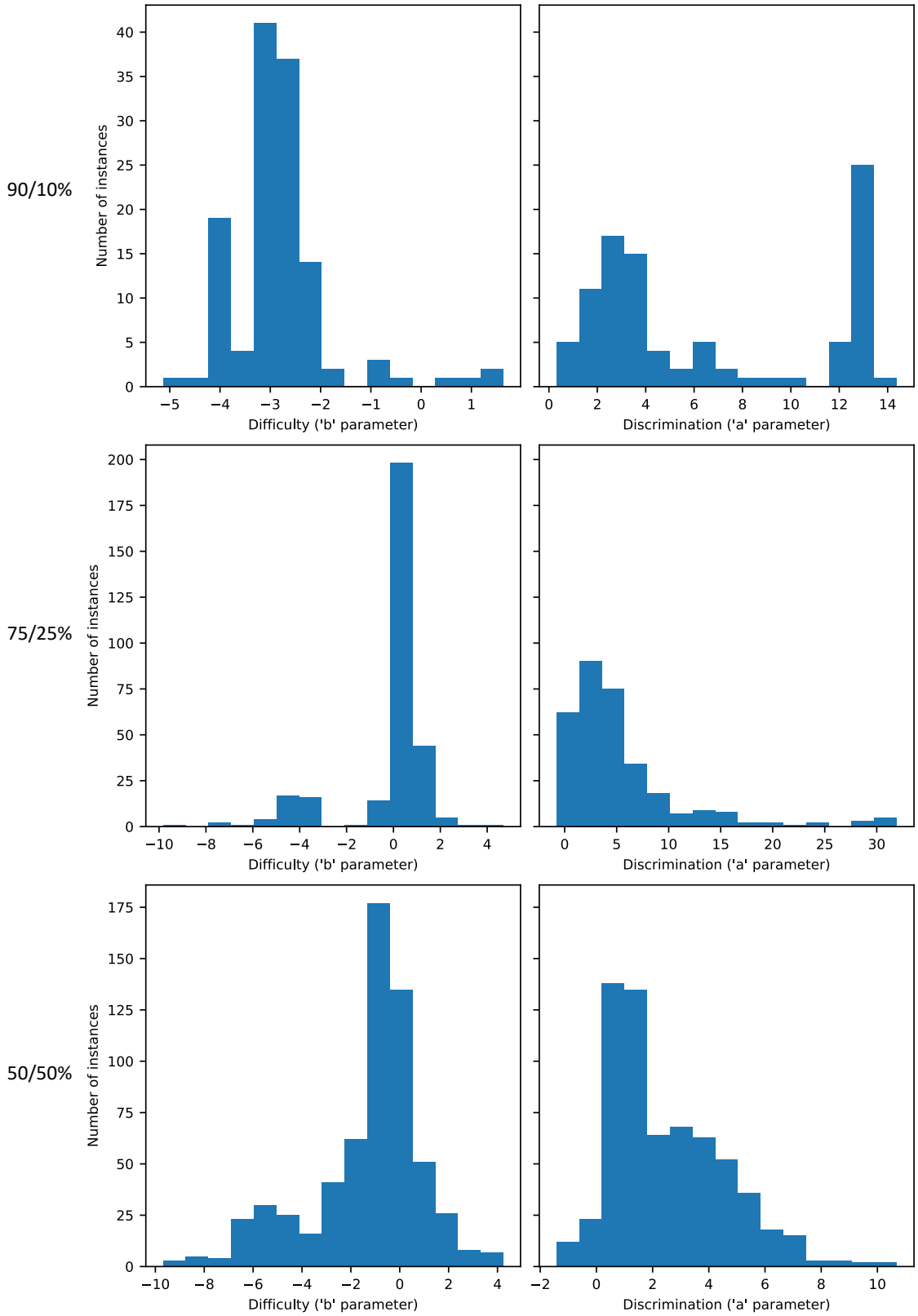

Figura 4.5. Visualização de dificuldade e discriminação de instâncias do conjunto de dados Árvores, para todas as proporções de treinamento e teste.

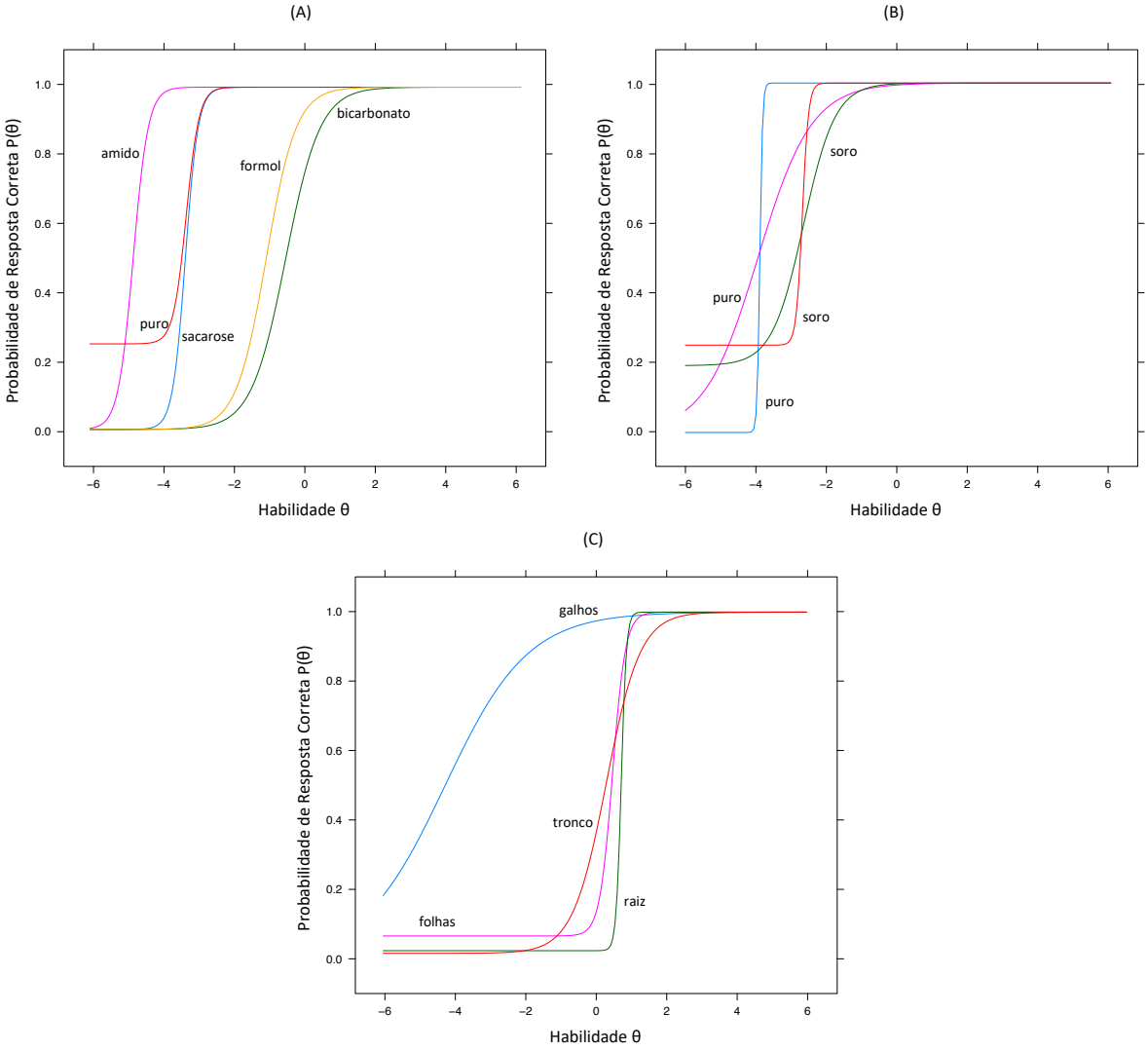

Figura 4.6. Exemplos aleatórios de curvas de característica de itens (ICC) de cada um dos conjuntos de dados: Adulterantes do Leite (A), Soro do Leite (B) e Árvores (C).

Leite, (B) Soro do Leite e (C) Árvores. Cada coluna representa o *fold* de treinamento e teste: 90/10%, 75/25% e 50/50%. A figura ainda mostra os classificadores artificiais otimista e pessimista nas bordas do gráfico e os classificadores aleatórios que produzem acurácia abaixo de 0.20 na linha (A), por volta de 0.50 na linha (B) e por volta de 0.25 na linha (C). O comportamento dos classificadores aleatórios é esperado, uma vez que eles produzem predições de amostras igualmente distribuídas, de forma que em (A) sua acurácia é aproximadamente 16%, pois há 6 classes possíveis ($1/6 \approx 0.1667$); em (B), a acurácia é por volta de 50%, pois esta é uma classificação binária; e em (C) a acurácia é aproximadamente 25%, pois há 4 classes possíveis.

A Figura 4.7 ainda mostra que, em geral, quanto maior a habilidade estimada pela

abordagem IRT, mais predições corretas o modelo treinado irá produzir. Além disso, quanto menor a proporção de dados de treinamento, mais modelos com habilidades inferiores estarão presentes. Por fim, pode-se ver que os problemas nos diferentes conjuntos de dados têm dificuldades inerentes distintas. Na Figura 4.7-(A), os modelos são mais condensados em níveis de acurácia maiores e habilidades entre -2 e 2 nos *folds* 90/10% e 75/25%. Já no *fold* 50/50%, os níveis de habilidade tendem a ter um maior impacto na acurácia. Na Figura 4.7-(B), os modelos são menos concentrados e suas variações de habilidades são maiores nos primeiros dois *folds* e semelhantes ao conjunto de dados anterior no último *fold*. Na Figura 4.7-(C), os modelos são muito mais espalhados, o que significa que modelos com baixas habilidades têm muito mais impacto na acurácia, diferente do que ocorre nos outros conjuntos de dados.

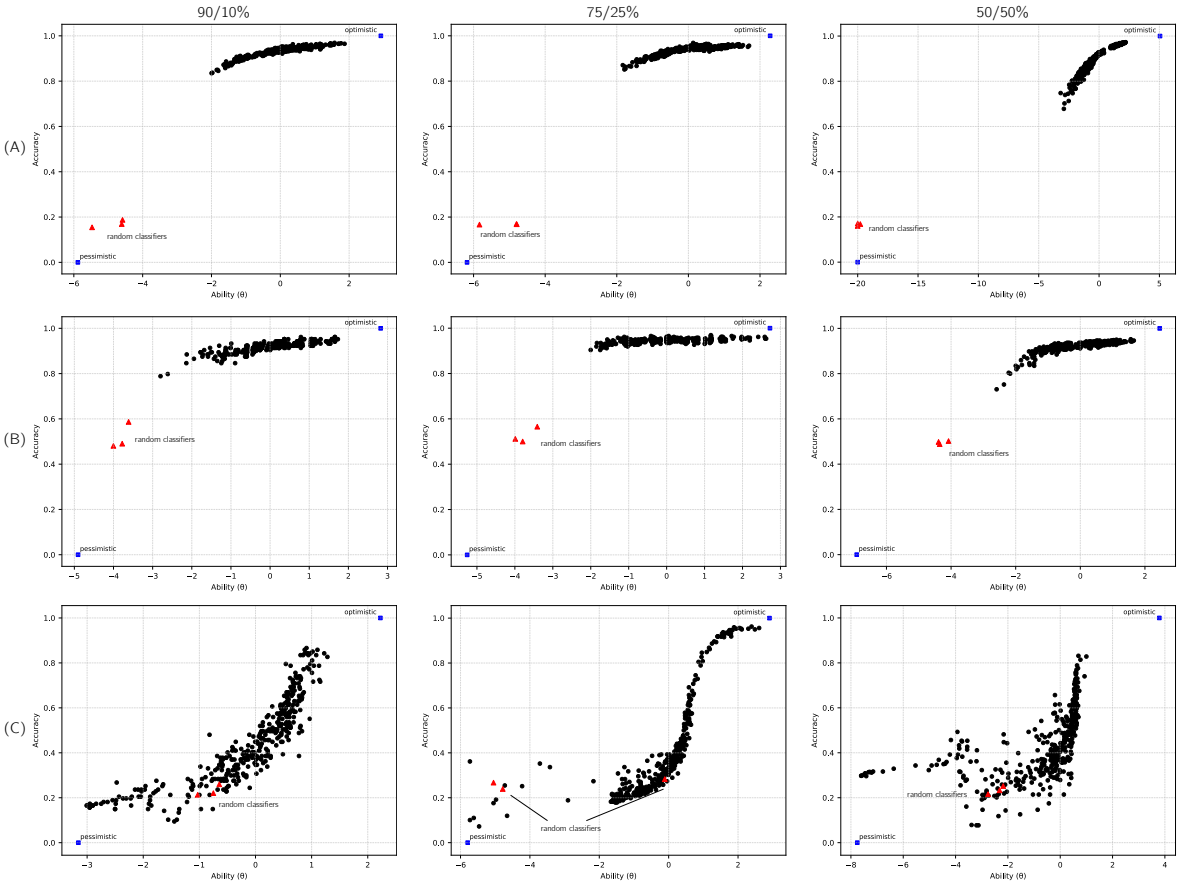

Figura 4.7. Gráfico de relacionamento entre o parâmetro de habilidade (θ) e a acurácia de classificação para cada conjunto de dados: (A) Adulterantes do Leite, (B) Soro do Leite e (C) Árvores; e para cada *fold* de treinamento e teste (90/10%, 75/25% e 50/50%). Os classificadores otimista e pessimista são representados por quadrados e os classificadores aleatórios são representados por triângulos. Em geral, quanto maior a habilidade de um modelo, maior é sua acurácia de classificação.

4.3.2 Resultados dos experimentos

A metodologia proposta foi aplicada para cada conjunto de dados, para proporção de treinamento e teste, o que gerou uma visualização de execução dos modelos em cada grupo de dificuldade ou discriminação. As Figuras 4.8, 4.9 e 4.10 representam as execuções dos modelos para as proporções de 90/10%, 75/25% e 50/50% de treinamento e teste, respectivamente. Nessas figuras, as linhas (A), (B) e (C) representam os conjuntos de dados utilizados, enquanto as duas colunas mostram um histograma para os parâmetros dificuldade e discriminação, respectivamente. Nos histogramas, em cada barra, pode-se visualizar o desempenho do modelo que tem o nível de habilidade compatível com as instâncias pertencentes àquele grupo.

Considerando grupos de dificuldade, pode-se ver que, em geral, o desempenho diminui no último grupo de dificuldade, o que é esperado, já que este grupo contém apenas as instâncias mais difíceis. Algumas acurácias mais baixas são encontradas nos grupos de instâncias mais fáceis (por exemplo, $\approx 0.95\%$ nos conjuntos de dados Adulterantes do Leite e Árvores). Em ambos os casos, os grupos restantes continuam apresentando altos valores de acurácia. Considerando a discriminação, o primeiro grupo em todos os conjuntos de dados apresentam valores mais baixos de acurácia, por volta de 0.80%, enquanto todos os outros grupos apresentam altos valores de acurácia. Especificamente na Figura 4.8-(C), pode-se visualizar que o método foi capaz de classificar corretamente todas as instâncias, de todos os grupos de dificuldade e discriminação.

4.3.3 Métodos de benchmark

Os experimentos com a metodologia NASirt foram realizados considerando ambos os grupos de dificuldade e discriminação, seguindo todos os passos descritos anteriormente na Seção 4.2.2. Além disso, foram realizados testes utilizando os mesmos n modelos considerados no *ranking* baseado em IRT, executando-os individualmente e contabilizando as predições corretas como um todo utilizando a abordagem de *majority-voting*. Resultados com altos valores de acurácia indicam uma boa seleção de modelos no *ranking* e, como o valor de n é pequeno, a votação realizada apenas com os n modelos é preferida em relação a todos os modelos treinados na coleção inicial.

Então, três valores diferentes compõem os resultados da metodologia NASirt proposta: etapas da metodologia com grupos de dificuldade e grupos de discriminação e *majority-voting* com modelos do *ranking* de IRT. Esses resultados foram comparados com resultados de outros métodos considerados *benchmark*: um modelo específico de CNN, uma abordagem de *majority-voting* geral e uma execução da ferramenta Auto-Keras.

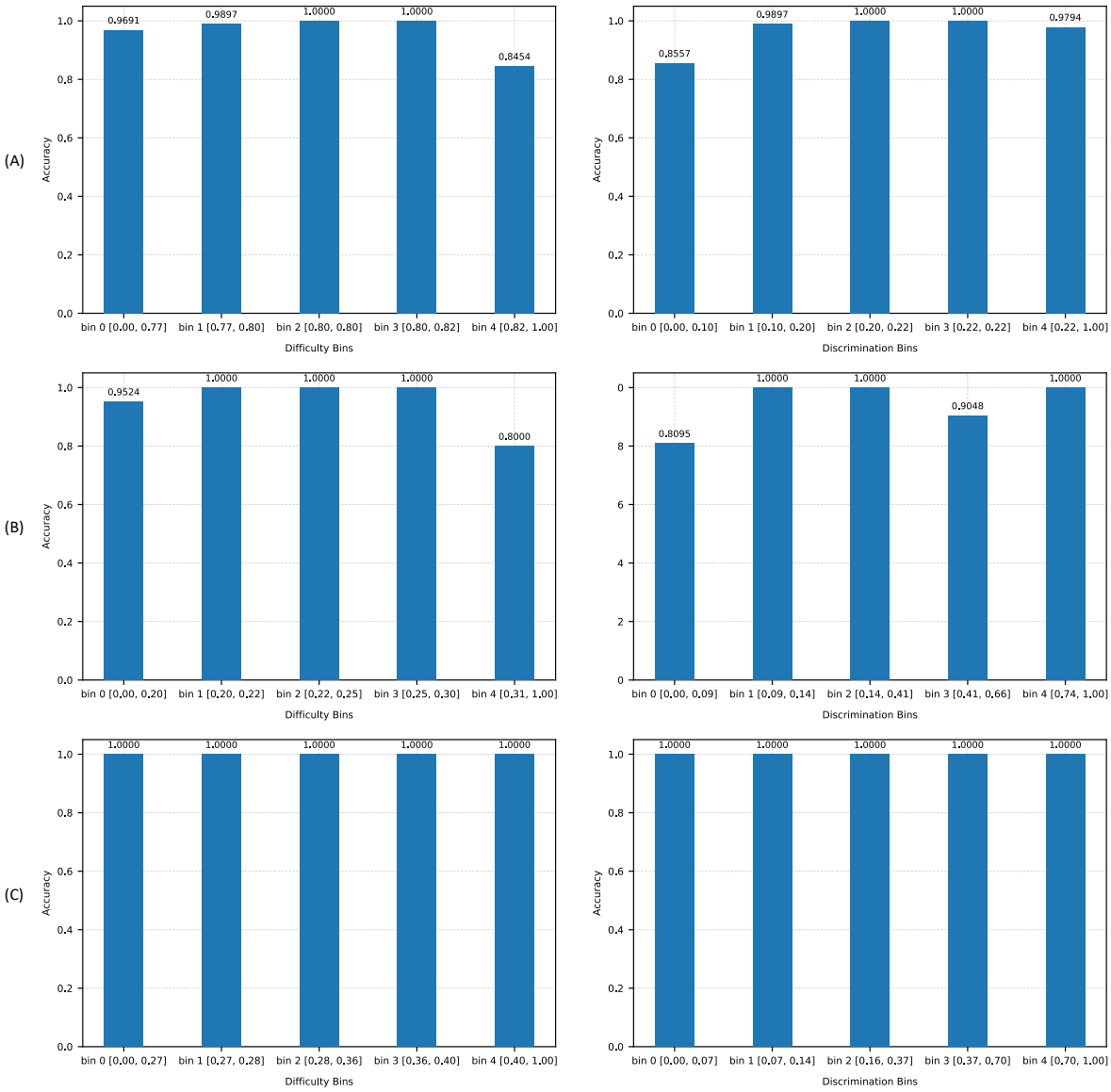

Figura 4.8. Gráfico de barras para a metodologia proposta considerando grupos de dificuldade e discriminação, usando a proporção 90/10% de treinamento e teste para os conjuntos de dados: (A) Adulterantes do Leite, (B) Soro do Leite e (C) Árvores. Cada barra representa a acurácia de classificação (eixo y) de um modelo que classifica apenas instâncias que estão no nível de dificuldade ou discriminação associado com a barra (descrito no eixo x). Acima de cada barra apresenta-se a acurácia exata daquele modelo em específico.

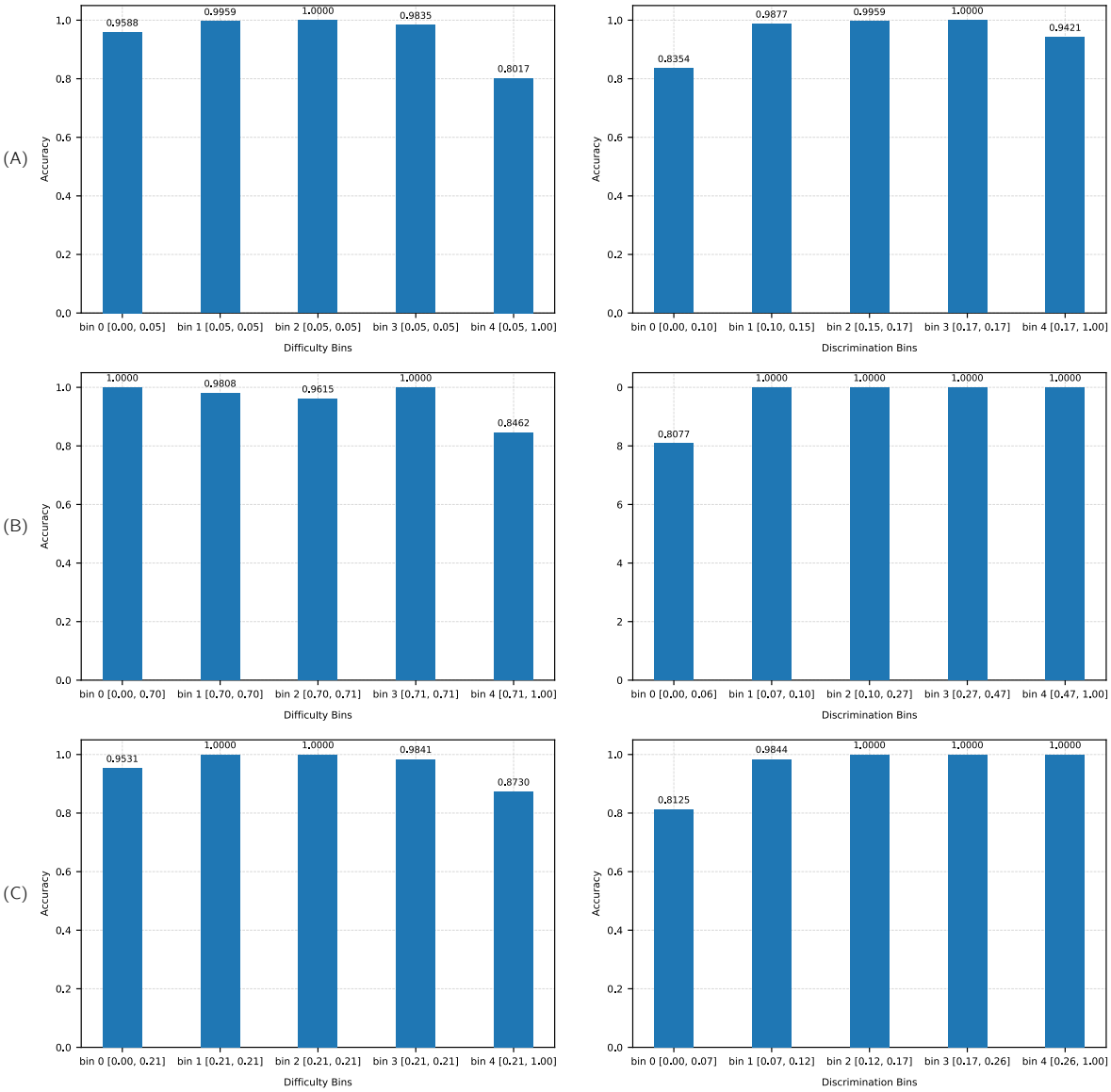

Figura 4.9. Gráfico de barras para a metodologia proposta considerando grupos de dificuldade e discriminação, usando a proporção 75/25% de treinamento e teste para os conjuntos de dados: (A) Adulterantes do Leite, (B) Soro do Leite e (C) Árvores. Cada barra representa a acurácia de classificação (eixo y) de um modelo que classifica apenas instâncias que estão no nível de dificuldade ou discriminação associado com a barra (descrito no eixo x). Acima de cada barra apresenta-se a acurácia exata daquele modelo em específico.

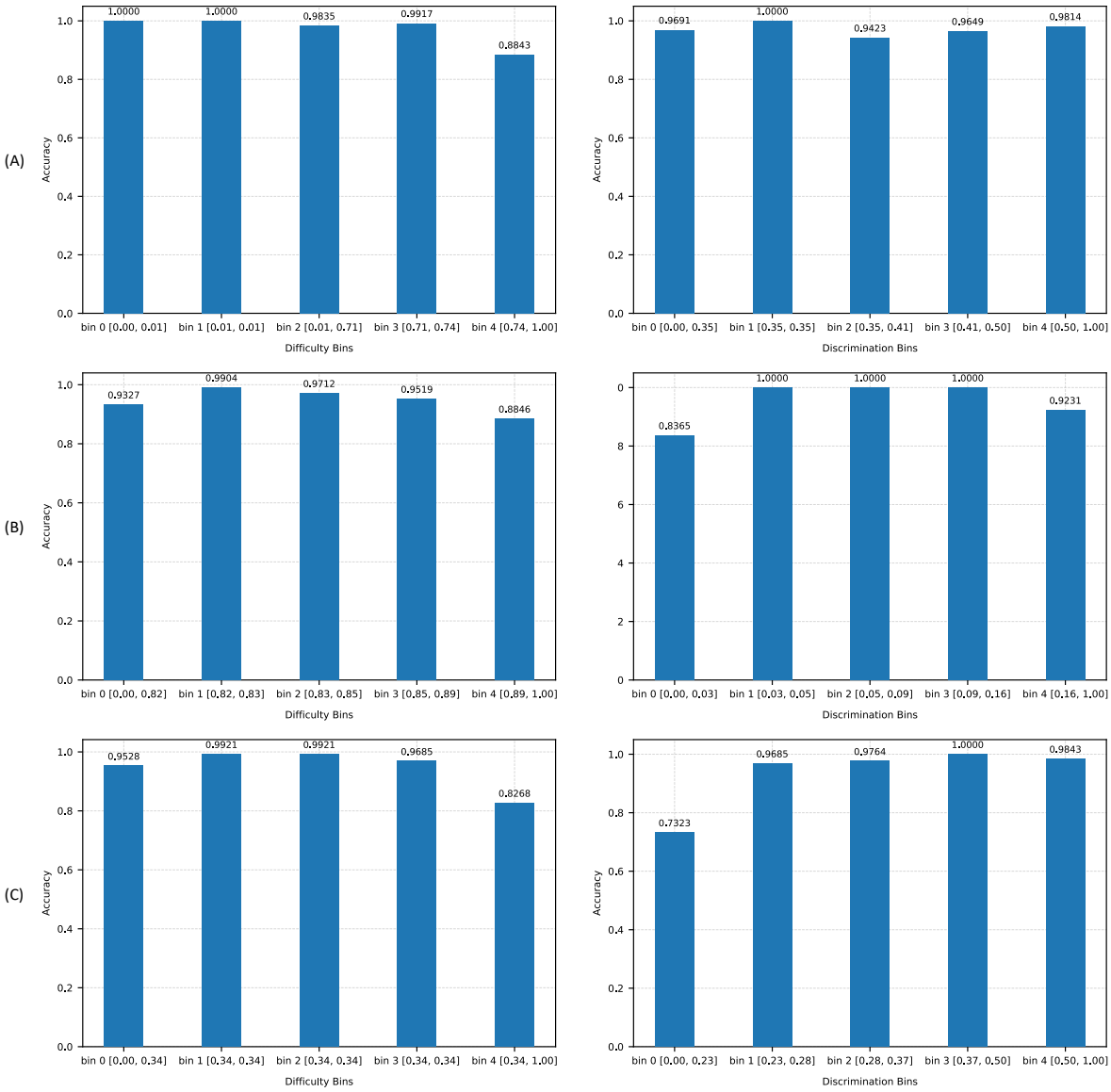

Figura 4.10. Gráfico de barras para a metodologia proposta considerando grupos de dificuldade e discriminação, usando a proporção 50/50% de treinamento e teste para os conjuntos de dados: (A) Adulterantes do Leite, (B) Soro do Leite e (C) Árvores. Cada barra representa a acurácia de classificação (eixo y) de um modelo que classifica apenas instâncias que estão no nível de dificuldade ou discriminação associado com a barra (descrito no eixo x). Acima de cada barra apresenta-se a acurácia exata daquele modelo em específico.

O modelo único de CNN utilizado como *benchmark* tem uma arquitetura relativamente simples, mas que obteve previamente os melhores desempenhos em dados espectrais do leite bovino, conforme apresentado no Capítulo 3. A arquitetura desta CNN compreende uma camada convolucional que aprende 32 filtros com tamanho de *kernel* 5, bem como uma camada densa com 1024 neurônios. Camadas de ativações utilizando LeakyReLU foram acrescentadas para adicionar não linearidade ao modelo [Maas et al., 2013] e operações de *dropout* foram utilizadas para prevenir a ocorrência de *overfitting* [Srivastava et al., 2014].

Já na abordagem de votação, os resultados são obtidos a partir das classificações de todos os m modelos treinados na coleção inicial de CNN e, por ser uma votação da maioria, a predição final da abordagem, para cada instância, é a predição mais frequente dessa instância considerando as predições de todos os modelos.

Por fim, a biblioteca Auto-Keras foi executada com as configurações originais para a busca de arquitetura de modelos utilizando o recurso denominado "AutoModel", que realiza o treinamento de modelos de CNN com variações dos seguintes hiperparâmetros: taxa de *dropout*, taxa de aprendizagem, otimizador, número de camadas convolucionais e densas, tamanho do *kernel*, número de neurônios em cada camada, e o uso ou não de normalização em lote (*batch normalization*).

O Auto-Keras executa diferentes "tentativas" (*trials*) de arquiteturas neurais e utiliza a técnica de morfismo de arquitetura, descrito anteriormente na Seção 4.1.3 [Wei et al., 2016; Jin et al., 2019], a fim de obter versões incrementais das arquiteturas. Nos experimentos realizados, o número de tentativas definido para executar o Auto-Keras foi 384, o mesmo número de modelos treinados na metodologia NASirt.

4.3.4 Comparação de resultados

As Tabelas 4.3, 4.4 e 4.5 apresentam uma comparação de acurácias para os conjuntos de dados Adulterantes do Leite, Soro do Leite e Árvores, respectivamente. Em cada tabela, as primeiras três colunas apresentam as execuções que são parte da metodologia NASirt: as colunas Dificuldade IRT e Discriminação IRT referem-se às etapas propostas na Seção 4.2.2 utilizando os parâmetros de IRT dificuldade e discriminação, enquanto a terceira coluna refere-se à votação da maioria com modelos do *ranking* de IRT definido pela metodologia. As outras três colunas das tabelas referem-se aos diferentes métodos usados como *benchmark*: o modelo de CNN único, a abordagem de votação considerando todos os modelos gerados na coleção de CNN e a execução do Auto-Keras.

Os valores de acurácia obtidos demonstram que a metodologia NASirt proposta

pode superar os métodos utilizados como *benckmark* na maioria dos casos. Para o conjunto de dados Adulterantes do Leite (Tabela 4.3), na média, todos os métodos apresentaram desempenho semelhante, com acurácias por volta de 96%, mas os valores obtidos pelo método proposto são ligeiramente superiores. Os resultados da metodologia proposta são melhores especificamente para a proporção 50/50%, o que indica que a metodologia é menos sensível a um conjunto de treinamento menor. Para o conjunto de dados Soro do Leite (Tabela 4.4), a metodologia proposta considerando ambos os parâmetros dificuldade e discriminação apresentaram o melhor desempenho geral. A votação com os modelos do *ranking* apresentaram valores de acurácia intermediários, seguidos pelos valores obtidos pelos métodos de *benchmark*. Por fim, para o conjunto de dados Árvores (Tabela 4.5), todos os métodos apresentaram comportamento semelhante, produzindo resultados mais precisos com conjuntos de treinamento maiores. No entanto, os métodos de *benchmark* são claramente mais sensíveis ao tamanho do conjunto de treinamento. Além disso, ao se comparar as duas abordagens de votação (considerando apenas os modelos do *ranking* e todos os modelos) os resultados destacam que a seleção de modelos realizada com base na IRT não apenas reduz drasticamente o número de modelos necessários, mas também oferece um aumento significativo de desempenho.

Tabela 4.3. Comparação dos resultados de acurácia para o conjunto de dados Adulterantes do Leite considerando a metodologia proposta e métodos de *benchmark*. As colunas Dificuldade IRT e Discriminação IRT representam os resultados da metodologia proposta, enquanto a coluna Votação (*ranking* IRT) representa o cálculo da votação da maioria considerando os modelos selecionados no Passo 5 da metodologia. A coluna CNN representa o resultado da arquitetura de CNN proposta no Capítulo 3. A coluna Votação representa a votação da maioria considerando todos as arquiteturas geradas no Passo 1 da metodologia. Por fim, a coluna Auto-Keras representa o resultado da execução da ferramenta Auto-Keras.

	Metodologia NASirt			*Benchmark*		
Fold	Dificuldade IRT	Discriminação IRT	Votação (*ranking* IRT)	CNN	Votação	Auto-Keras
90/10%	0.9680	0.9649	0.9649	0.9608	0.9526	0.9567
75/25%	0.9480	0.9521	0.9505	0.9685	0.9554	0.9537
50/50%	0.9719	0.9715	0.9728	0.9538	0.9596	0.9442
Média	0.9626	0.9628	0.9627	0.9610	0.9558	0.9515

A Tabela 4.6 apresenta os valores de precisão obtidos pela execução do método NASirt para cada proporção de treinamento e teste e para cada conjunto de dados, considerando os parâmetros de IRT dificuldade (b) e discriminação (a). Esses valo-

Tabela 4.4. Comparação dos resultados de acurácia para o conjunto de dados Soro do Leite considerando a metodologia proposta e métodos de *benchmark*. As colunas Dificuldade IRT e Discriminação IRT representam os resultados da metodologia proposta, enquanto a coluna Votação (*ranking* IRT) representa o cálculo da votação da maioria considerando os modelos selecionados no Passo 5 da metodologia. A coluna CNN representa o resultado da arquitetura de CNN proposta no Capítulo 3. A coluna Votação representa a votação da maioria considerando todos as arquiteturas geradas no Passo 1 da metodologia. Por fim, a coluna Auto-Keras representa o resultado da execução da ferramenta Auto-Keras.

	Metodologia NASirt			*Benchmark*		
Fold	Dificuldade IRT	Discriminação IRT	Votação (*ranking* IRT)	CNN	Votação	Auto-Keras
90/10%	0.9519	0.9423	0.9423	0.9231	0.9231	0.9038
75/25%	0.9577	0.9615	0.9577	0.9462	0.9500	0.9423
50/50%	0.9462	0.9519	0.9442	0.9365	0.9346	0.9230
Média	0.9519	0.9519	0.9480	0.9353	0.9359	0.9230

Tabela 4.5. Comparação dos resultados de acurácia para o conjunto de dados Árvores considerando a metodologia proposta e métodos de *benchmark*. As colunas Dificuldade IRT e Discriminação IRT representam os resultados da metodologia proposta, enquanto a coluna Votação (*ranking* IRT) representa o cálculo da votação da maioria considerando os modelos selecionados no Passo 5 da metodologia. A coluna CNN representa o resultado da arquitetura de CNN proposta no Capítulo 3. A coluna Votação representa a votação da maioria considerando todos as arquiteturas geradas no Passo 1 da metodologia. Por fim, a coluna Auto-Keras representa o resultado da execução da ferramenta Auto-Keras.

	Metodologia NASirt			*Benchmark*		
Fold	Dificuldade IRT	Discriminação IRT	Votação (*ranking* IRT)	CNN	Votação	Auto-Keras
90/10%	1.0000	1.0000	1.0000	0.9764	0.9843	0.9527
75/25%	0.9623	0.9591	0.9560	0.9340	0.7701	0.8993
50/50%	0.9465	0.9323	0.9417	0.4425	0.3176	0.9259
Média	0.9696	0.9638	0.9659	0.7843	0.6906	0.9181

res representam o número de verdadeiros positivos dividido pelo total das predições positivas (verdadeiros positivos e falsos positivos). Em outras palavras, os valores da tabela representam a precisão do método ao predizer, das das instâncias preditas como positivas, quantas delas são realmente positivas.

Tabela 4.6. Valores da métrica de precisão da execução da metodologia NASirt com as variações de parâmetro de IRT dificuldade (b) e discriminação (a), para cada proporção de treinamento e teste, e para cada conjunto de dados analisado: Adulterantes, Soro do Leite e Árvores.

	Adulterantes		Soro do Leite		Árvores	
Fold	NASirt (b)	NASirt (a)	NASirt (b)	NASirt (a)	NASirt (b)	NASirt (a)
90/10%	0.9815	0.9882	0.9743	0.9452	1.0000	0.9911
75/25%	0.9699	0.9666	0.9403	0.9474	0.9736	0.9758
50/50%	0.9656	0.9643	0.9306	0.9398	0.9065	0.9155
Média	0.9723	0.9730	0.9484	0.9441	0.9600	0.9608

4.3.5 Complexidade de modelos

A fim de oferecer uma visão geral da complexidade do método proposto e dos modelos que fizeram parte dos experimentos conduzidos, pode-se calcular o número de parâmetros de cada rede neural. Esses parâmetros, em gera, são considerados os pesos dos neurônios, que são ajustados (aprendidos) durante o treinamento da rede. Esses pesos contribuem para o poder preditivo dos modelos e a contagem total de parâmetros pode ser considerada uma medida de complexidade para o modelo.

Nos experimentos conduzidos, o método NASirt proposto é composto por um *ranking* de modelos CNN que obtiveram os melhores valores de *true score* em IRT. O número de modelos nesse *ranking* é arbitrário, e a sugestão da metodologia é que seja definido um valor baixo (por exemplo, 5). Então, pode-se entender a complexidade da metodologia proposta como a soma do número de parâmetros (pesos) em cada modelo de rede neural que o compõe.

Já para os métodos utilizados como *benchmark*, onde foram utilizados um modelo de CNN único, uma abordagem de votação e o Auto-Keras, no primeiro caso, por se tratar de um modelo único, a complexidade é simplesmente o número de parâmetros da rede neural utilizada. Para a abordagem de votação, a complexidade considerada é a soma dos parâmetros de todos os modelos individuais considerados. O Auto-Keras, por outro lado, utiliza a técnica de morfismo de redes neurais, gerando assim um único modelo final com um número reduzido de parâmetros.

A contagem dos parâmetros de rede é apresentada na Tabela 4.7. Nesta tabela, denota-se uma comparação quantitativa das complexidades da metodologia proposta e dos métodos de *benchmark*. A tabela permite concluir que a seleção de um *ranking* de IRT produz uma complexidade geral mais perto do modelo de CNN único e do Auto-Keras, e essas complexidades são muito inferiores aos valores apresentados pela votação com todos os modelos.

Os experimentos foram executados no ambiente Google Colab, que fornece acesso virtualizado a uma CPU Intel Xeon de 2,3Ghz, uma GPU Nvidia Tesla K80 com 12GB de memória GDDR5 e 13GB de memória RAM. Os tempos de execução foram armazenados para comparar o desempenho dos métodos NASirt e Auto-Keras. Apenas os métodos NASirt e Auto-Keras foram comparados, pois os resultados de votação são basicamente uma contagem das predições realizadas nas execuções existentes, e o modelo de CNN tem seu desempenho apresentado no Capítulo 3. A variação de dificuldade ou discriminação da metodologia NASirt pode reaproveitar uma única execução, então os tempos se referem a ambos os parâmetros.

Para o conjunto Adulterantes do Leite, o maior tempo de execução, considerando as diferentes proporções de treinamento e teste, foi de aproximadamente 6 horas e 43 minutos (NASirt) e 10 horas e 11 minutos (Auto-Keras). Para o conjunto Soro do Leite, o maior tempo foi de aproximadamente 5 horas e 20 minutos (NASirt) e 9 horas e 51 minutos (Auto-Keras). Por fim, para o conjunto Árvores, o maior tempo foi de 6 horas e 10 minutos (NASirt) e 11 horas e 27 minutos (Auto-Keras).

Tabela 4.7. Número de parâmetros de rede para os modelos do método NASirt e para os métodos de *benchmark*: modelo de CNN, votação e Auto-Keras.

Conjunto de dados	NASirt	Modelo CNN	Votação	Auto-Keras
Adulterantes do Leite	46.8M	30.6M	3.7B	20.1M
Soro do Leite	107.4M	30.6M	3.7B	20.9M
Árvores	346.7M	69.7M	8.2B	72.1M

Os valores são referentes ao maior número de parâmetros encontrado entre todas as proporções de treinamento e tese, para cada conjunto de dados. M = milhões; B = Bilhões de parâmetros.

Ao analisar a complexidade dos métodos, o NASirt apresenta o número de parâmetros equivalente à soma dos parâmetros dos modelos utilizados no *ranking*. Como, nos experimentos, o número de modelos do *ranking* foi definido como 5, os valores da coluna NASirt da Tabela 4.7 representam a soma dos parâmetros de 5 redes neurais distintas. O método Auto-Keras, por possuir uma abordagem de morfismo de rede, possui a complexidade representada apenas pelo seu modelo final. Em outras pa-

lavras, o valor da complexidade apresentado na coluna Auto-Keras não considera cada variação de arquitetura realizada pelo método, e a complexidade é calculada apenas para a última arquitetura gerada. É possível notar que o Auto-Keras realiza mais trabalho pelo seu tempo de execução descrito anteriormente, todos maiores que o NASirt.

4.4 Discussão

Este capítulo apresentou a metodologia de AutoML denominada NASirt, que possibilita gerar automaticamente arquiteturas de redes neurais adequadas para conjuntos de dados cujas instâncias são amostras de análises espectrais. A metodologia se baseia na utilização da Teoria de Resposta ao Item, que oferece um entendimento da dificuldade e das discriminações das instâncias, bem como uma noção de qualidade dos modelos gerados, através de conceitos que analisam as habilidades desses modelos.

Nos experimentos realizados neste capítulo, diferentes conjuntos de dados espectrais foram submetidos à metodologia proposta e os resultados foram comparados com outros métodos de *benchmark*. Três métodos analisados compõem a metodologia NASirt: a execução das etapas considerando o parâmetro dificuldade, considerando o parâmetro discriminação e a execução de uma abordagem *majority-voting* com os modelos filtrados pelo IRT nas etapas do método.

Os resultados obtidos pelos experimentos demonstram que o NASirt não só é capaz de gerar, na média, o maior número de acertos nas classificações, mas também é capaz de caracterizar instâncias com níveis de dificuldades e discriminação. Além disso, a definição de modelos mais ou menos habilidosos é uma característica inovadora da metodologia proposta. Esses aspectos são possíveis graças ao emprego da IRT.

A metodologia proposta permeia dois tópicos que vêm se tornando muito populares nos últimos anos na área de Aprendizado de Máquina: *Fairness* e *Explainable Artificial Intelligence* (XAI). *Fairness*, ou "justiça", estuda a transparência dos algoritmos e vieses nos conjuntos de dados. Um algoritmo de aprendizado de máquina é dito "justo" se os seus resultados são independentes de variáveis consideradas sensíveis [Mehrabi et al., 2019]. Normalmente, o tema *fairness* está relacionado a implicações legais das decisões tomadas por algoritmos de aprendizado de máquina, como preconceito ou favoritismo em relação a um indivíduo ou grupo com base em suas características inerentes ou adquiridas. O XAI refere-se a métodos de aprendizado de máquina que possam fornecer explicações sobre sua lógica e suas tomadas de decisão de forma que o método possa ser compreendido facilmente. Essas explicações são importantes para garantir que o algoritmo seja, por sua vez, justo, identificando possíveis vieses e proble-

mas nos dados, além de garantir que o algoritmo execute como esperado [Gilpin et al., 2018].

A metodologia NASirt pode ser utilizada para apoiar estudos mais aprofundados sobre *fairness* e XAI com ajuda da avaliação realizada pelo modelo de IRT para o *ranking* de modelos proposto. A utilização de IRT é essencial para oferecer um método capaz de realizar uma explicação local do processo de tomada de decisões, considerando cada instância individualmente. A metodologia proposta permite explorar a explicabilidade dos modelos, avaliando dificuldade, discriminação e habilidade de modelos, e descrevendo o raciocínio por trás das decisões do método.

Capítulo 5

Conclusão e trabalhos futuros

Dados de coordenadas espectrais que representam materiais biológicos complexos são utilizados com frequência para o estudo de seus componentes a fim de se obter características presentes nas amostras. Os processos e metodologias apresentados neste trabalho envolvem a análise das coordenadas espectrais através de métodos computacionais avançados, capazes de oferecer o reconhecimento de padrões precisos a partir dessas amostras.

Neste trabalho, um conjunto de dados espectrais obtidos de uma coleção de amostras do leite bovino foi submetido a uma metodologia que utiliza a tarefa de classificação para detectar amostras comprometidas com diferentes tipos de adulterantes. As técnicas empregadas são capazes de oferecer desempenhos com precisões interessantes e a arquitetura de rede neural convolucional proposta ofereceu médias de acerto de 97,38%, para a detecção da presença ou ausência de adulterantes no leite, e de 96,13% para a detecção específica de qual dos adulterantes conhecidos foi encontrado nas amostras. Os resultados foram comparados com outros métodos, inclusive com métodos que se baseiam no processamento realizado internamente pelo equipamento de espectroscopia, que utiliza diversas calibrações e conhecimentos prévios das características do leite. Em geral, a rede neural utilizada sem qualquer preprocessamento e sem conhecimentos prévios das amostras é capaz de oferecer resultados melhores do que os outros métodos estudados.

Os métodos de aprendizado de máquina, especialmente os baseados em redes neurais artificiais, carecem de uma fácil explicabilidade. Em geral, não há uma relação direta entre as arquiteturas desses modelos e a função que eles executam. Muitas vezes, os modelos são referidos como sendo "caixas-pretas", dadas suas interpretações complexas. A Teoria de Resposta ao Item abordada no trabalho oferece técnicas para melhorar a explicabilidade dos modelos: pode-se obter características de complexidade

das instâncias analisadas, como dificuldade e discriminação, bem como as habilidades inerentes aos modelos. Por exemplo, instâncias mais difíceis podem ser classificadas corretamente por modelos mais habilidosos, mas podem ser um problema para modelos com menores níveis de habilidade.

As características da IRT permitiram o desenvolvimento de uma metodologia inovadora de aprendizado de máquina automatizado, denominada NASirt. O método envolve a obtenção de arquiteturas de redes neurais convolucionais de forma automatizada e, além disso, pode explicar as habilidades dos modelos encontrados. Diversos experimentos foram conduzidos para avaliar o método NASirt, envolvendo diferentes conjuntos de dados. Os valores de acurácia foram considerados em diferentes proporções de treinamento e teste, para cada conjunto de dados e o método NASirt apresentou acurácias superiores em praticamente todas as execuções. O método proposto é ainda mais vantajoso ao se considerar conjuntos de treinamentos menores, o que confirma que a utilização de modelos mais habilidosos oferece maior desempenho geral. Além disso, diferentemente de métodos convencionais de AutoML, o NASirt ainda pode caracterizar a complexidade das instâncias dos conjuntos de dados analisados, bem como definir automaticamente o grupo de modelos mais habilidosos, cujas arquiteturas produzem classificações mais precisas.

5.1 Trabalhos futuros

Como continuidade do trabalho, algumas oportunidades de melhorias para o método proposto podem ser destacadas, como descrito a seguir.

- Utilizar a extração de *features* e saliências dos espectros, implementada no trabalho e descrita na Seção 3.5, Capítulo 3, página 45, para propor uma associação com o modelo de IRT utilizado no NASirt.
- Propor melhorias na estratégia de definição dos grupos de instâncias referentes aos parâmetros dificuldade e discriminação de IRT, realizada no Passo 6 da metodologia e apresentada na Seção 4.2.2, Capítulo 4, página 61.
- Propor melhorias ao NASirt relacionadas a diferentes técnicas de AutoML, como a utilização de estimativas de desempenho considerando o treinamento de baixa fidelidade, a fim de acelerar o treinamento de modelos e obter estimativas de habilidade de antemão [Elsken et al., 2019]. Outra técnica que pode ser empregada é o morfismo de rede, que visa minimizar o custo computacional de treinamento de modelos individuais, bem como permite diminuir o número final de parâmetros das redes, simplificando, portanto, os modelos.

- Explorar a explicabilidade dos modelos de forma a tornar o NASirt um método XAI.
- Generalizar o método para ser executado em outras aplicações além dos dados espectrais, como imagens bidimensionais.

Referências Bibliográficas

Abadi, M. et al. (2016). TensorFlow: A System for Large-scale Machine Learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pp. 265--283, Savannah, GA, USA. USENIX Association.

Alpaydin, E. (2014). *Introduction to Machine Learning*. The MIT Press, Cambridge, 3^a edição. ISBN 9780262028189.

Alves da Rocha, R.; Paiva, I. M.; Anjos, V.; Furtado, M. A. M. & Valenzuela, M. J. (2015). Quantification of whey in fluid milk using confocal raman microscopy and artificial neural network. *Journal of Dairy Science*, 98(6):3559--3567. ISSN 0022-0302.

Andrade, D. F.; Tavares, H. R. & Valle, R. C. (2000). *Teoria de Resposta ao Item: conceitos e aplicações*. Associação Brasileira de Estatística, São Paulo.

Arlot, S. & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40--79. ISSN 1935-7516.

Baker, F. B. (2001). *The Basics of Item Response Theory*. ERIC Clearinghouse on Assessment and Evaluation, College Park, 2 edição. ISBN 1-886047-03-0.

Baker, F. B. & Kim, S.-H. (2017). *The Basics of Item Response Theory Using R*. Springer International Publishing, Cham, 1 edição. ISBN 9783319542058.

Baker, M. J.; Trevisan, J.; Bassan, P.; Bhargava, R.; Butler, H. J.; Dorling, K. M.; Fielden, P. R.; Fogarty, S. W.; Fullwood, N. J.; Heys, K. A.; Hughes, C.; Lasch, P.; Martin-Hirsch, P. L.; Obinaju, B.; Sockalingum, G. D.; Sulé-Suso, J.; Strong, R. J.; Walsh, M. J.; Wood, B. R.; Gardner, P. & Martin, F. L. (2014). Using Fourier transform IR spectroscopy to analyze biological materials. *Nature protocols*, 9(8):1771--1791. ISSN 1750-2799.

Bergstra, J. & Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305. ISSN 1532-4435.

Bhandare, A.; Bhide, M.; Gokhale, P. & Chandavarkar, R. (2016). Applications of Convolutional Neural Networks. *International Journal of Computer Science and Information Technologies*, 7(5):2206 – 2215. ISSN 0975-9646.

Botelho, B. G.; Reis, N.; Oliveira, L. S. & Sena, M. M. (2015). Development and analytical validation of a screening method for simultaneous detection of five adulterants in raw milk using mid-infrared spectroscopy and PLS-DA. *Food Chemistry*, 181:31 – 37. ISSN 0308-8146.

Cardoso, L. F. F.; Santos, V. C. A.; Francês, R. S. K.; Prudêncio, R. B. C. & Alves, R. C. O. (2020). Decoding machine learning benchmarks.

Chalmers, R. (2012). mirt: A Multidimensional Item Response Theory Package for the R Environment. *Journal of Statistical Software, Articles*, 48(6):1--29. ISSN 1548-7660.

Chen, T. & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785--794, New York, NY, USA. ACM.

Chollet, F. et al. (2015). Keras. Available at https://keras.io.

de Carvalho, B. M. A.; de Carvalho, L. M.; dos Reis Coimbra, J. S.; Minim, L. A.; de Souza Barcellos, E.; da Silva Júnior, W. F.; Detmann, E. & de Carvalho, G. G. P. (2015). Rapid detection of whey in milk powder samples by spectrophotometric and multivariate calibration. *Food Chemistry*, 174:1--7. ISSN 0308-8146.

Dong, X.; Yu, Z.; Cao, W.; Shi, Y. & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, 14(2):241--258. ISSN 2095-2236.

Elsken, T.; Metzen, J. H. & Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861 – 874. ISSN 0167-8655. ROC Analysis in Pattern Recognition.

Gilpin, L. H.; Bau, D.; Yuan, B. Z.; Bajwa, A.; Specter, M. & Kagal, L. (2018). Explaining Explanations: An Overview of Interpretability of Machine Learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 80–89.

Gondim, C. d. S.; Junqueira, R. G.; de Souza, S. V. C.; Ruisánchez, I. & Callao, M. P. (2017). Detection of several common adulterants in raw milk by MID-infrared spectroscopy and one-class and multi-class multivariate strategies. *Food Chemistry*, 230:68--75. ISSN 18737072.

Griffiths, P.; De Haseth, J. & Winefordner, J. (2007). *Fourier Transform Infrared Spectrometry*. Chemical Analysis: A Series of Monographs on Analytical Chemistry and Its Applications. Wiley, 2ª edição. ISBN 9780471194040.

Hastie, T.; Tibshirani, R. & Friedman, J. (2017). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2ª edição. ISBN 0387848576.

Hutter, F.; Kotthoff, L. & Vanschoren, J. (2019). *Automated Machine Learning*. Springer International Publishing, Cham, 1ª edição. ISBN 978-3-030-05318-5.

Ioffe, S. & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448--456, Lille, France. PMLR.

James, G.; Witten, D.; Hastie, T. & Tibshirani, R. (2017). *An Introduction to Statistical Learning: with Applications in R*. Springer, New York. ISBN 1461471370.

Jin, H.; Song, Q. & Hu, X. (2019). Auto-Keras: An Efficient Neural Architecture Search System. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, p. 1946–1956, New York, NY, USA. Association for Computing Machinery.

Kamal, M. & Karoui, R. (2015). Analytical methods coupled with chemometric tools for determining the authenticity and detecting the adulteration of dairy products: A review. *Trends in Food Science & Technology*, 46(1):27 – 48. ISSN 0924-2244.

Kartheek, M.; Anton Smith, A.; Kottai Muthu, A. & Manavalan, R. (2011). Determination of Adulterants in Food: A Review. *Journal of Chemical and Pharmaceutical Research*, 3(2):629 – 636.

Kasemsumran, S.; Thanapase, W. & Kiatsoonthon, A. (2007). Feasibility of near-infrared spectroscopy to detect and to quantify adulterants in cow milk. *Analytical Sciences*, 23(7):907–910.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q. & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Guyon,

I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S. & Garnett, R., editores, *Advances in Neural Information Processing Systems 30*, pp. 3146--3154. Curran Associates, Inc.

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pp. 1137--1143, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Kubat, M. (2017). *An introduction to machine learning*, volume 2. Springer, Cham. ISBN 9783319639130.

Kuhn, M. & Johnson, K. (2013). *Applied Predictive Modeling*. Springer-Verlag, New York, 1ª edição. ISBN 978-1-4614-6849-3.

Kyriakides, G. & Margaritis, K. (2020). An Introduction to Neural Architecture Search for Convolutional Networks.

Lalor, J. P.; Wu, H.; Munkhdalai, T. & Yu, H. (2018). Understanding deep learning performance through an examination of test set difficulty: A psychometric case study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4711--4716, Brussels, Belgium. Association for Computational Linguistics.

Lalor, J. P.; Wu, H. & Yu, H. (2016). Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 648--657, Austin, Texas. Association for Computational Linguistics.

Lipton, Z. C. (2018). The mythos of model interpretability. *Queue*, 16(3):30:31--30:57. ISSN 1542-7730.

Liu, J.; Osadchy, M.; Ashton, L.; Foster, M.; Solomon, C. J. & Gibson, S. J. (2017). Deep convolutional neural networks for Raman spectrum recognition: A unified solution. *Analyst*, 142(21):4067--4074. ISSN 13645528.

Luna, A. S.; Pinho, J. S. A. & Machado, L. C. (2016). Discrimination of adulterants in uht milk samples by nirs coupled with supervision discrimination techniques. *Anal. Methods*, 8:7204–7208.

Maas, A. L.; Hannun, A. Y. & Ng, A. Y. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, USA.

Maimon, O. & Rokach, L., editores (2010). *Data Mining and Knowledge Discovery Handbook*. Springer, New York, 2ª edição. ISBN 978-0-387-09822-7.

Marqués, A.; García, V. & Sánchez, J. (2012). Exploring the behaviour of base classifiers in credit scoring ensembles. *Expert Systems with Applications*, 39(11):10244 – 10250. ISSN 0957-4174.

Martínez-Plumed, F.; Prudêncio, R. B.; Martínez-Usó, A. & Hernández-Orallo, J. (2019). Item response theory in AI: Analysing machine learning classifiers at the instance level. *Artificial Intelligence*, 271:18 – 42. ISSN 0004-3702.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K. & Galstyan, A. (2019). A survey on bias and fairness in machine learning.

Nicolaou, N.; Xu, Y. & Goodacre, R. (2010). Fourier transform infrared spectroscopy and multivariate analysis for the detection and quantification of different milk species. *Journal of Dairy Science*, 93(12):5651–5660.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825--2830.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45. ISSN 1531-636X.

Polikar, R. (2012). Ensemble learning. In Zhang, C. & Ma, Y., editores, *Ensemble Machine Learning*, pp. 1–34. Springer, Boston, MA.

Ramirez, J. A.; Posada, J. M.; Handa, I. T.; Hoch, G.; Vohland, M.; Messier, C. & Reu, B. (2015). Near-infrared spectroscopy (NIRS) predicts non-structural carbohydrate concentrations in different tissue types of a broad range of tree species. *Methods in Ecology and Evolution*, 6(9):1018–1025.

Russell, S. & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Pearson, Edinburgh Gate, 3ª edição. ISBN 0136042597.

Santos, P. M.; Pereira-Filho, E. R. & Rodriguez-Saona, L. E. (2013). Rapid detection and quantification of milk adulteration using infrared microspectroscopy and chemometrics analysis. *Food Chemistry*, 138(1):19--24. ISSN 03088146.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85--117. ISSN 0893-6080.

Simonyan, K.; Vedaldi, A. & Zisserman, A. (2013). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv e-prints*, p. arXiv:1312.6034.

Smith, M. R.; Martinez, T. & Giraud-Carrier, C. (2014). An instance level analysis of data complexity. *Machine Learning*, 95(2):225--256. ISSN 1573-0565.

Souza, S. S.; Cruz, A. G.; Walter, E. H.; Faria, J. A.; Celeghini, R. M.; Ferreira, M. M.; Granato, D. & de S. Sant'Ana, A. (2011). Monitoring the authenticity of Brazilian UHT milk: A chemometric approach. *Food Chemistry*, 124(2):692 – 695. ISSN 0308-8146.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929--1958.

Stuart, B. H. (2012). *Infrared Spectroscopy of Biological Applications: An Overview*. American Cancer Society.

Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning: an introduction*. The MIT Press, Cambridge, 2^a edição. ISBN 9780262039246.

Torgo, L. & Gama, J. (1996). Regression by classification. In Borges, D. L. & Kaestner, C. A. A., editores, *Advances in Artificial Intelligence*, pp. 51--60, Berlin, Heidelberg. Springer Berlin Heidelberg.

van der Linden, W. J. & Hambleton, R. K. (1997). *Handbook of Modern Item Response Theory*. Springer-Verlag, New York, 1 edição. ISBN 9781475726916.

Wei, T.; Wang, C.; Rui, Y. & Chen, C. W. (2016). Network Morphism. In Balcan, M. F. & Weinberger, K. Q., editores, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 564--572, New York, New York, USA. PMLR.

Witten, I.; Frank, E.; Hall, M. & Pal, C. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Cambridge, 4ª edição. ISBN 9780128042915.

Zeiler, M. D. & Fergus, R. (2014). Visualizing and understanding convolutional networks. In Fleet, D.; Pajdla, T.; Schiele, B. & Tuytelaars, T., editores, *Computer Vision – ECCV 2014. Lecture Notes in Computer Science*, volume 8689, pp. 818--833, Cham. Springer International Publishing.

İmamoğlu, N.; Zhang, C.; Shmoda, W.; Fang, Y. & Shi, B. (2017). Saliency detection by forward and backward cues in deep-CNN. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 430–434, Beijing, China. IEEE. ISSN 2381-8549.