Universidade Federal do Rio de Janeiro
Centro de Ciências Matemáticas e da Natureza
Observatório do Valongo

Medindo catálogo de redshifts fotométricos com o LSST Data Preview 0.2: um estudo de ponta a ponta

Andreia Araujo Dourado

Rio de Janeiro
Setembro de 2025

Medindo catálogo de redshift fotométricos nos dados simulados do LSST Data Preview 0.2: um estudo de ponta a ponta.

Andreia Araujo Dourado

Trabalho de Conclusão de Curso submetido ao Observatório do Valongo, Universidade Federal do Rio de Janeiro, como requisito necessário para a obtenção do título de Astrônomo.

Orientadores: Bruno Azevedo Lemos Moraes, Julia de Figueiredo Gschwend

Rio de Janeiro
Setembro de 2025

CIP - Catalogação na Publicação

A739m

Araujo Dourado, Andreia
Medindo catálogo de redshifts fotométricos com o LSST Data Preview 0.2: um estudo de ponta a ponta / Andreia Araujo Dourado. -- Rio de Janeiro, 2025.
97 f.

Orientador: Bruno Azevedo Lemos Moraes.
Coorientadora: Julia de Figueiredo Gschwend.
Trabalho de conclusão de curso (graduação) - Universidade Federal do Rio de Janeiro, Observatório do Valongo, Bacharel em Astronomia, 2025.

1. Cosmologia Observacional. 2. Redshift fotométrico. 3. Machine Learning. I. Azevedo Lemos Moraes, Bruno, orient. II. de Figueiredo Gschwend, Julia, coorient. III. Título.

Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a), sob a responsabilidade de Miguel Romeu Amorim Neto - CRB-7/6283.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
CCMN - OBSERVATÓRIO DO VALONGO
DEPARTAMENTO DE ASTRONOMIA

PROJETO FINAL

RELATÓRIO DA COMISSÃO JULGADORA

ALUNA: Andreia Araújo Dourado (DRE 115030443)

TÍTULO DO TRABALHO: "MEDINDO CATÁLOGOS DE REDSHIFTS FOTOMÉTRICOS COM O LSST DATA PREVIEW 0.2: UM ESTUDO DE PONTA A PONTA"

DATA DA DEFESA: 19 de setembro de 2025 às 10:00 h

MEMBROS DA COMISSÃO JULGADORA:

Prof.ª Karín Menéndez-Delmestre – Presidente – OV/UFRJ

Prof. Bruno Moraes – Orientador - IF/UFRJ

Dr.ª Julia Gschwend – Co-orientadora - LIneA

Prof. Ribamar Reis – IF/UFRJ

CANDIDATA: Andreia Araujo Dourado

Andreia Araújo Dourado

Rio de Janeiro, 19 de setembro de 2025.

Prof. Carlos Roberto Rabaça
Vice-coord. de Grad. do Curso de Astronomia

Ladeira do Pedro do Antônio, 43 – Saúde – 20080-090 – tel.: 3938-0544 – e-mail: sec.academ@ov.ufrj.br

À minha mãe (in memoriam). Das poucas lembranças que tenho, a do quanto ela sonhava com esse momento é uma das mais vivas.

Agradecimentos

Agradeço primeiramente à minha ancestralidade, que me manteve de pé e guiou meu caminho até aqui, mesmo quando eu nem achava que chegaria. A todos os orixás, exus, pombagiras, malandros, ciganos, caboclos, pretos velhos e erês que me acompanharam nessa jornada, em especial aos que carrego comigo, nunca terei palavras para descrever o quanto sou grata. Ao meu pai Ogun, que me deu o empurrão inicial para que eu tomasse o caminho certo, e à minha guardiã, Dona Maria Mulambo, que nunca me deixou esquecer onde eu queria chegar. Agradeço também a todos que possibilitaram esse reencontro com minha espiritualidade.

Agradeço imensamente à minha família. Desde que eu falei pela primeira vez que queria fazer Astronomia, mesmo sem entenderem direito que rumo exatamente eu tomaria com essa profissão, nunca mediram esforços para que isso fosse possível. E, mesmo com o diploma não vindo com o passar dos anos, nunca deixaram de me apoiar e compreender as dificuldades pelas quais passei. As lembranças periódicas para que eu não desistisse foram um pilar fundamental para que essa formatura fosse possível. Amo muito vocês.

Aos meus amigos do ensino médio: Amanda, Laura, Cintia, João, Estevam, Fabrício, que estiveram presentes à distância. Obrigada por todo apoio e por deixarem as coisas mais leves sempre que paramos para conversar, como se ainda estivéssemos em algum corredor do IFMA rindo por uma tarde inteira.

Aos amigos do ensino médio que estiveram presencialmente: Allanis e Adones. Não consigo montar uma combinação de dupla mais peculiar que essa para ter por perto após todos esses anos (quem conhece sabe...), e sou imensamente grata por ter vocês aqui. Obrigada por tudo que vocês já fizeram por mim e por atenuarem a saudade de casa sempre que estamos juntos.

Aos meus amigos Ariel e João: das coisas boas que levo dessa graduação, vocês com certeza estão no topo da lista. Sou muito grata e feliz pela amizade verdadeira que construímos ao longo dos anos. Eu nem sei o que seria de mim sem vocês ao meu lado no Valongo e fora dele. Muito obrigada por tudo.

Aos meus amigos Gama, Sarah e Pelou: muito obrigada pelo apoio e amizade de vocês, por torcerem e vibrarem por mim de perto a cada conquista. E por ainda me aceitarem na mesa do Village, mesmo depois das mil e uma suecas que eu quebrei pra conseguir terminar esse texto.

Ao meu namorado Wallace, muito obrigada por ter sido meu ponto de equilíbrio e refúgio incontáveis vezes. Pela compreensão nos meus momentos de caos e por acreditar

em mim mais do que eu mesma muitas vezes. Por me demonstrar amor também em todas as vezes que me ajudou a não desistir do meu propósito, ou todas as vezes que fala orgulhoso para alguém sobre o que eu estudo. E por todas as pessoas que trouxe para minha vida, as quais também sou muito grata por me acolherem e me apoiarem.

Aos meus amigos de pesquisa: Iago, muito obrigada por toda a ajuda com este trabalho, por ter segurado minha mão milhares de vezes, pelos desabafos, fofocas e bandejões; Boni, muito obrigada por estar comigo nos últimos tempos, pela ajuda nas matérias e pelo alívio cômico nos momentos de desespero; e Amanda, pelas trocas de conhecimento que muito me ajudaram na escrita.

Agradeço ao projeto OBA! Planetário Itinerante, pela oportunidade de, ao divulgar astronomia e astronáutica nas escolas, me reconectar com a Andreia do passado que decidiu ser astrônoma através da OBA. Ser planetarista me trouxe de volta o amor pela Astronomia, a certeza de que estou na profissão certa e ainda reforça todos os dias o meu propósito enquanto cientista. Muito além de trabalho, é sankofa, imani e nia em suas essências para mim. Obrigada a toda equipe por compartilharem seus conhecimentos, aprendo muito com todos vocês. E aos amigos que fiz, muito obrigada por todos os jogos, memes, risadas, nosso dialeto e o apoio de vocês. E um agradecimento mais que especial à promessa que fizeram comigo pela quinta bolsa!

Também reservo uma boa parte dos meus agradecimentos a todos os professores que me ajudaram a construir esse caminho. Ao professor Ronivaldo Pacheco, meu professor de física do ensino médio, que me apresentou um novo mundo de conhecimento nas áreas de astronomia e astronáutica, através do nosso grupo de competição na MOBFOG. Ao professor Fabio Sales, meu orientador no PIBIC/EM com um projeto de IC em divulgação de astronomia e astronáutica na minha escola, projeto esse que me fez escolher esse curso de graduação. Ao professor Adrian Colucci, que foi meu orientador acadêmico todos esses anos e sempre levou a sério esse papel, não tenho palavras para agradecer toda a atenção e compreensão que sempre teve, e por ter sido um dos principais responsáveis pela minha permanência na universidade. À professora Diana Andrade, minha primeira orientadora de IC, que me auxiliou dentro e fora da universidade, muito obrigada por me guiar novamente ao amor pela ciência e por tudo que fez por mim. Ao professor Bruno Morgado, que foi meu professor nas disciplinas da ênfase computacional, pelas excelentes aulas e por me apresentar o uso de machine learning na astronomia, que foi o ponto inicial para o projeto desenvolvido neste trabalho.

Agradeço imensamente a toda equipe do LIneA, pelo empenho em manter sempre o bom funcionamento dos ambientes de trabalho e aprimorá-los de acordo com as necessidades dos grupos de pesquisa, permitindo a produção científica de qualidade. Em

especial, agradeço à Helosia, ao Luigi e ao Cristiano, com quem trabalhei diretamente, por serem sempre muito solícitos e me permitirem aprender tanto com vocês.

Aos meus orientadores: Julia Gschwend, muito obrigada por todos os ensinamentos passados e pelas oportunidades que me foram dadas. Para além da orientação neste trabalho, levo para vida uma referência de profissional nas áreas que mais amo, sou muito grata pela oportunidade de trabalhar com você; Bruno Moraes, muito obrigada por aceitar orientar meu TCC e por estruturar esse projeto que me abriu tantas portas que eu jamais imaginei, tendo sempre o cuidado de observar minhas limitações. Muito obrigada pelas orientações de trabalho e de vida, pela compreensão que sempre teve comigo e por me ajudar a construir o caminho profissional que quero seguir. Agradeço imensamente aos dois por fazerem parte da construção desse momento tão importante para mim.

Agradeço ao CNPq, pelo apoio financeiro que permitiu o desenvolvimento deste projeto, através do INCT do e-Universo.

Agradeço aos professores Karín Menéndez-Delmestre e Ribamar Rondon, por aceitarem fazer parte da banca examinadora deste trabalho.

A todos que de alguma forma, direta ou indiretamente, me ajudaram a chegar até aqui, muito obrigada. Mesmo que não citados diretamente ou não façam mais parte da minha vida, sou grata por tudo que fizeram por mim.

E, por fim, eu agradeço a mim mesma. Só eu sei a aventura que foi sair de casa para fazer essa graduação. E, ainda assim, eu consegui :).

"A educação é um elemento importante na luta pelos direitos humanos. É o meio para ajudar os nossos filhos e as pessoas a redescobrirem a sua identidade e, assim, aumentar o seu autorespeito. Educação é o nosso passaporte para o futuro, pois o amanhã só pertence ao povo que prepara o hoje."

Malcolm X

Resumo

Medindo catálogo de redshifts fotométricos com o LSST Data Preview 0.2: um estudo de ponta a ponta.

Andreia Araujo Dourado

Orientadores: Bruno Azevedo Lemos Moraes, Julia de Figueiredo Gschwend

Resumo do Trabalho de Conclusão de Curso submetido ao Observatório do Valongo, Universidade Federal do Rio de Janeiro, como requisito necessário para a obtenção do título de Astrônomo.

A estimativa de redshifts fotométricos (photo-z) é uma tarefa fundamental e desafiadora na astronomia observacional moderna, especialmente para levantamentos de galáxias fotométricos em grande escala como o LSST (Legacy Survey of Space and Time). Algoritmos de machine learning têm se destacado como ferramentas promissoras, devido à sua flexibilidade e capacidade de modelar relações complexas e não lineares. Neste trabalho, exploramos as etapas para gerar um catálogo de photo-zs, com aplicação do algoritmo TPZ, baseado na construção de florestas aleatórias de árvores de decisão. Construímos um *training set* representativo com os dados simulados do LSST DP0.2, aplicando seleções de qualidade. Analisamos o impacto dos hiperparâmetros principais do TPZ e das diferentes escolhas de atributos para treinamento do modelo no resultado final. Mostramos as métricas de viés, dispersão e fração de outliers para os valores previstos de photo-z, além das métricas para avaliar a calibração das PDFs geradas, obtendo resultados consistentes e satisfatórios para o objetivo do nosso estudo. Aplicamos o modelo gerado no conjunto de dados de alta volumetria, através do pipeline pz-compute, desenvolvido pelo LIneA (Laboratório Interinstitucional de e-Astronomia) como parte de sua contribuição para o LSST. Construímos a distribuição de galáxias por redshift da tabela final, e analisamos a performance do algoritmo no cluster de computadores.

palavras chave: *astronomia, cosmologia observacional, redshift fotométrico, machine learning*

Rio de Janeiro
Setembro de 2025

Abstract

Measuring photometric redshifts with LSST Data Preview 0.2: an end-to-end study

Andreia Araujo Dourado

Advisors: Bruno Azevedo Lemos Moraes, Julia de Figueiredo Gschwend

Abstract submitted to the Valongo Observatory, Federal University of Rio de Janeiro, in fulfillment of the requirements for the degree of Astronomer.

Photometric redshift (photo-z) estimation is a fundamental and challenging task in modern observational astronomy, especially for large-scale surveys like the LSST. Machine learning algorithms have proven to be promising tools due to their flexibility and ability to model complex, non-linear relationships. In this work, we describe the steps to generate a photo-z catalog using the TPZ algorithm, which is based on building random forests of decision trees. We constructed a representative training set using simulated data from LSST DP0.2, applying quality selections. We analyzed the influence of TPZ's main hyperparameters and different sets of training features on the final results. We evaluated the model using metrics such as bias, dispersion, outlier fraction, and calibration indicators for the predicted probability density functions (PDFs), obtaining results that were consistent and satisfactory for our study goals. The trained model was applied to a high-volume dataset using the pz-compute pipeline, developed by LIneA as part of its contribution to the LSST. We built the redshift distribution of galaxies from the final output table and assessed the algorithm's performance on the computing cluster.

keywords: *astronomy, observational cosmology, photometric redshift , machine learning*

Rio de Janeiro
September 2025

Lista de Figuras

Lista de Tabelas

Abreviações

DP 0.2	**D**ata **P**review **0.2**
DR	**D**ata **R**elease
ML	**M**achine **L**earning
LSST	**L**egacy **S**urvey of **S**pace and **T**ime
LIneA	**L**aboratório **In**terinstitucional de **e-A**stronomia
IDAC	**I**ndependent **D**ata **A**cess **C**enter
DC2	**D**ata **C**hallenge **2**
RAIL	**R**edshift **A**ssessment **I**nfrastructure **L**ayers
TPZ	**T**rees for **P**hoto -**Z**
CDM	**C**old for **D**ark -**M**atter
DESC	**D**ark **E**nergy **S**cience **C**olaboration
DM	**D**ata for **M**anagement
DESI	**D**ark **E**nergy **S**pectroscopic **I**nstrument
RSP	**R**ubin **S**cience **P**latform

Sumário

Capítulo 1

Introdução

A cosmologia no séc. XXI caminha lado a lado aos significativos avanços tecnológicos, e seu impacto na qualidade da instrumentação observacional, da modelagem teórica e da capacidade computacional que se tem disponível para fazer ciência. Questões fundamentais como a natureza da energia escura e da matéria escura, entre outras, motivam uma série de investigações que levam ao desenvolvimento de grandes levantamentos cosmológicos, com o objetivo de mapear a estrutura em grande escala do universo e testar os fundamentos do modelo cosmológico padrão (ΛCDM). Dentre os grandes projetos nos últimos anos, podemos citar Sloan Digital Sky Survey (SDSS) (Margony, 1999), Dark Energy Survey (DES)(Dark Energy Survey Collaboration et al., 2016), Dark Energy Spectroscopic Instrument (DESI) (Collaboration et al., 2022), Euclid (ESA) (Mellier et al., 2025), Legacy Survey of Space and Time (LSST) (Ivezić, 2019).

Um dos objetivos centrais da cosmologia é a medição das propriedades fundamentais do universo a partir da distribuição espacial de galáxias. Para isto, é necessário conhecer suas distâncias radiais em relação ao nosso referencial, o que é possível através da medição do efeito Doppler causado pela expansão do Universo, gerando um desvio para o vermelho (redshift) na luz das galáxias que podemos medir a partir das imagens captadas pelos telescópios (Ryden, 2017).

Para garantir que as inferências feitas nos valores dos parâmetros cosmológicos sejam confiáveis, é necessário que as medidas de redshift sejam robustas. Este é um grande desafio para os levantamentos fotométricos modernos: garantir que viéses nas medidas de redshift não impactem negativamente nas medidas desses parâmetros.

Uma forma robusta de se fazer essa medida é através da espectroscopia, técnica na qual se obtém um espectro com as linhas de emissão e absorção da galáxia que pode ser comparado com medições feitas na Terra para calcular o desvio para o vermelho nas mesmas, chamado de redshift espectroscópico (spec-z). Apesar de ser uma técnica muito precisa, fazer levantamentos de galáxias com espectroscopia traz algumas limitações, por ser extremamente custoso, além da capacidade reduzida de observações afetar objetos de magnitude mais fraca e com distribuição espectrais de energia (SEDs) sem propriedades marcantes (Soo, 2018), o que causa um viés na amostragem do diagrama de cor-magnitude. Esse viés é crítico em cosmologia, pois distorce as medidas de aglomeramento de galáxias e pode comprometer a análise da estrutura em escala do Universo. Como alternativa, sugere-se o uso da fotometria das galáxias para estimar o redshift (Baum, 1962), chamado de redshfit fotométrico (photo-z), de modo que são utilizados os fluxos das galáxias medidos em filtros, geralmente de banda larga, que fornecem um espectro de baixa resolução. Essa perda de informação no espectro traz um aumento nas incertezas da medição do photo-z em comparação ao spec-z; por outro lado, o aumento na capacidade de observação compensa estatisticamente devido ao grande número e variedade de objetos obtidos por fotometria.

Dentre as formas de se obter o photo-z, temos dois grupos: os métodos de ajuste de modelos, onde é feito um ajuste com espectros prévios para obtenção do redshift, e os métodos de modelos de treinamento, que utilizam equações empíricas que relacionam os valores de magnitudes, derivados dos fluxos, com o valor de redshift, construídas com base em amostras espectroscópicas. Dentro dos métodos empíricos está o uso de machine learning (ML).

ML é uma área da inteligência artificial na qual os algoritmos desenvolvidos são capazes de identificar padrões em dados e fazer previsões ou classificações com base nesses padrões, sem serem explicitamente programados para cada tarefa específica (Marsland, 2015). No contexto do cálculo de photo-z, o avanço tecnológico e a crescente demanda de dados de grandes levantamentos astronômicos fazem o uso do ML uma ferramenta poderosa para essa estimativa, principalmente por sua capacidade de aprender a relação complexa entre as cores das galáxias e seus redshifts com base em um conjunto de treinamento com spec-zs conhecidos.

Dentre os diversos tipos de algoritmos de ML disponíveis, um dos mais populares

são as árvores de decisão. Se caracterizam pela divisão hierárquica dos dados com base em condições lógicas simples, até que se chegue a uma predição no nó final da árvore (Željko Ivezić & Gray, 2020). Esse tipo de modelo é intuitivo, robusto a outliers e capaz de capturar interações não-lineares entre variáveis, características valiosas no contexto de estimativa de photo-zs. A escolha por usar árvores de decisão neste trabalho está diretamente relacionada à sua eficiência, interpretabilidade e à possibilidade de formar métodos de ensemble, como florestas aleatórias, que combinam várias árvores para melhorar a generalização e a robustez do modelo.

Como aplicação das florestas aleatórias de árvores de decisão, temos o algoritmo TPZ (do inglês, Trees for Photo-Z), proposto por Carrasco Kind & Brunner (2013), especificamente para predição de photo-zs. O TPZ vai além da simples predição pontual: ele estima a função de densidade de probabilidade (PDF) para o redshift de cada objeto, fornecendo uma estimativa probabilística que é crucial para análises cosmológicas robustas.

Os dados utilizados neste trabalho são dados de simulação preliminares do LSST Rubin Observatory Legacy Survey of Space and Time (Rubin LSST), um dos maiores levantamentos astronômicos planejados para os próximos anos. Com observações previstas ao longo de uma década, o LSST cobrirá cerca de 18000 graus quadrados do hemisfério sul, gerando um grande volume de dados em seis bandas fotométricas no óptico (ugrizy) (Ivezić, 2019). Diante desse cenário, torna-se essencial o desenvolvimento e a validação de ferramentas capazes de lidar com essa escala de informação.

Como parte da preparação para os dados reais, foi disponibilizado o Data Preview 0.2 (DP0.2), baseado na simulação Data Challenge 2 (DC2) desenvolvida pelo LSST Dark Energy Science Collaboration (LSST DESC) et al. (2021). O DC2 parte de simulações cosmológicas N-body e avança até a geração de imagens realistas nos padrões LSST, seguidas por observações simuladas completas; o DP0.2 é um reprocessamento das imagens do DC2 com uma versão atualizada dos pipelines do LSST. Esses dados simulados reproduzem de forma realista as observações esperadas no LSST, permitindo à comunidade testar algoritmos, treinar modelos e estabelecer pipelines que futuramente serão aplicados aos dados observacionais. Assim, o DP0.2 cumpre um papel estratégico no amadurecimento das técnicas e infraestruturas computacionais que serão utilizadas no levantamento real.

Nesse contexto, destaca-se o desenvolvimento do RAIL (Redshift Assessment Infrastructure Layers), uma plataforma que visa padronizar a execução e a avaliação de diferentes métodos de estimação de redshifts fotométricos (photo-z) (Team et al., 2025). O RAIL oferece uma estrutura modular e extensível, na qual diversos algoritmos podem ser integrados de forma transparente, facilitando análises comparativas e reprodutíveis. Há vários algoritmos de photo-z implementados no RAIL, dentre eles uma versão atualizada para Python 3 do TPZ.

Idealizado no contexto da colaboração científica *Dark Energy Science Collaboration* (DESC), o RAIL foi posteriormente incorporado aos *pipelines* do departamento de *Data Management* (DM) do LSST para produzir estimativas de photo-z de aplicação genérica que farão parte dos catálogos oficiais liberados pelo LSST a cada *Data Release* (DR). Além destas, também serão disponibilizadas tabelas federadas [1] (integradas aos dados fotométricos no banco de dados central do projeto) produzidas por colaboradores internacionais com estimativas geradas por outros algoritmos, ampliando a diversidade de aplicações em casos de uso científicos.

Como parte do programa de contribuições *in-kind* [2], o Brasil possui um centro de dados local do LSST, conectado a uma rede internacional de centros, os chamados *Independent Data Access Centers* (IDACs). O Laboratório Interinstitucional de e-Astronomia (LIneA), que hospeda o IDAC-Brasil, será o responsável por entregar esses catálogos alternativos de photo-z anualmente. Para cumprir essa tarefa, foi desenvolvido o *pipeline* `pz-compute`, que atua como uma camada de suporte para incorporar a estrutura do RAIL ao ambiente de computação de alto desempenho do IDAC (LIneA (Laboratório Interinstitucional de e-Astronomia), 2025).

Diante deste cenário, o presente trabalho tem como objetivo fazer um estudo de ponta a ponta da geração de um catálogo de redshifts fotométricos em um conjunto de dados simulados, utilizando o algoritmo TPZ, entendendo como as diferentes etapas da geração de um catálogo de redshifts fotométricos impactam a performance final do modelo. Os testes realizados neste trabalho e descritos nos próximos capítulos contribuíram para a validação da instalação e dos resultados científicos gerados pelo `pz-compute`, parte

[1] Tabelas de dados externas, fornecidas por usuários e geradas por parceiros, combinadas aos dados do catálogo primário.

[2] Apoios não financeiros em troca de direito de acesso aos dados.

importante das atividades de preparação para a operação do IDAC, que se iniciará junto com a primeira liberação de dados do LSST.

O texto a seguir está estruturado da seguinte forma: no Capítulo 2, é apresentada a fundamentação teórica sobre redshift fotométrico, ML e o tipo de algoritmo utilizado. O Capítulo 3 descreve as etapas realizadas para a estimativa dos redshifts fotométricos, desde a obtenção dos dados até a geração do catálogo final. No Capítulo 4, são detalhados os dados utilizados e todo o tratamento aplicado para a construção do *training set* final. O Capítulo 5 apresenta os resultados obtidos com o algoritmo TPZ, incluindo a análise das métricas de desempenho e a discussão sobre o comportamento do modelo. Por fim, o Capítulo 6 traz as conclusões do estudo, destacando as principais contribuições, limitações e sugestões para trabalhos futuros.

Capítulo 2

Redshift fotométrico e Machine Learning

A Cosmologia é um ramo da Astronomia que estuda o Universo em larga escala, buscando entender desde seu surgimento e como se deu seu evolução (Harrison, 2020). Sua construção teórica moderna começou no início do século XX, com a aplicação da recém-formulada relatividade geral ao universo como um todo. Em 1917, Albert Einstein propôs o primeiro modelo cosmológico relativístico apoiado às ideias de Ernst Mach, assumindo um universo estático e homogêneo. No mesmo ano, Willem de Sitter apresentou uma solução alternativa, que já sugeria o afastamento das galáxias — o chamado "efeito de Sitter". Poucos anos depois, Aleksandr Friedmann propôs modelos cosmológicos em expansão, rompendo com a ideia de um universo estático. Paralelamente, observações espectroscópicas feitas por Vesto Slipher, seguidas por análises de Carl Wirtz e outros, já indicavam que muitas galáxias apresentavam desvios para o vermelho. Essas pistas teóricas e observacionais culminaram nos trabalhos de Edwin Hubble no final da década de 1920, que estabeleceu uma relação aproximadamente linear entre a velocidade de afastamento das galáxias e sua distância — a Lei de Hubble-Lemaître (Waga, 2005).

Essa relação fornece evidência direta da expansão do universo e fundamenta a interpretação do redshift cosmológico como um efeito do próprio alongamento do espaço, conceito de grande importância para a cosmologia observacional.

2.1 O Universo em expansão e o Redshift Cosmológico

Quando observamos em escala cosmológica, um sinal luminoso emitido por um objeto muito distante quando observado na Terra apresenta um redshift. Isso porque a onda eletromagnética viaja pelo Universo em expansão e tem, então, seu comprimento de onda aumentado, causando o desvio para o vermelho no espectro (Lambourne, 2010).

Para grandes escalas, assume-se pelo princípio cosmológico que o Universo está em uma expansão homogênea e isotrópica, o que implica que, ao comparar as distâncias entre os objetos ao longo do tempo, é possível relacioná-las por um fator que mantém essas propriedades (Ryden, 2017). Esse é o chamado fator de escala $a(t)$, que caracteriza a expansão do universo dependente do tempo. Em outras palavras, o fator de escala mede quantas vezes as distâncias físicas eram menores no passado, quando comparadas com as mesmas medidas , como ilustrado na Figura 2.1.

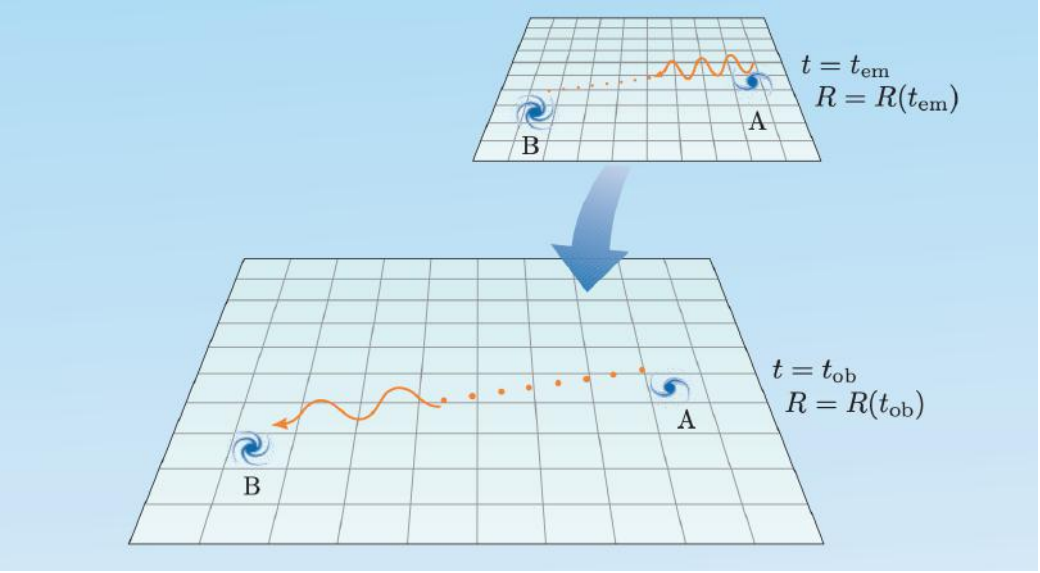


FIGURA 2.1. Ilustração do redshift como consequência da expansão do Universo (Lambourne, 2010).

Matematicamente, esse espaço é descrito pela métrica Friedmann-Robertson-Walker (FRW) [A], que é escrita como

$$ds^2 = -c^2dt^2 + a^2(t)[\frac{1}{1-\frac{kr^2}{R^2}}dr^2 + r^2(d\theta^2 + sen^2\theta d\phi^2)], \tag{2.1}$$

onde ds^2 é o elemento de intervalo espaço-temporal, c é a velocidade da luz no vácuo, R é o raio de curvatura do espaço, k é o parâmetro de curvatura espacial e $a(t)$ é o fator de escala, que acompanha o termo espacial da equação.

Uma vez que a luz é a principal fonte de informação do Universo, o redshift é um parâmetro que pode ser medido diretamente e a partir dele pode-se obter informações importantes acerca da expansão do universo, ao relacioná-lo com o fator de escala.

Tomando, por exemplo, dois pontos r_1 e r no espaço descrito, fixando r na origem do sistema de coordenadas e r_1 na direção radial, ou seja, com $d\theta = 0$ e $d\phi = 0$. Um sinal luminoso é emitido por uma galáxia situada em r_1 no instante t_1 e esse mesmo sinal é observado na Terra, situada em $r = 0$, no instante t_o. Dada a métrica que descreve o espaço e a definição quantitativa do redshift dada por

$$z = \frac{\lambda_o - \lambda_1}{\lambda_1}, \tag{2.2}$$

tem-se que

$$1 + z = \frac{a(t_o)}{a(t_1)}. \tag{2.3}$$

Esta é a relação entre o redshift e o fator de escala, cujo cálculo detalhado pode ser lido no Apêndice A. A partir dela, tendo o valor de redshift e convencionando $a(t_o) = 1$ para o tempo atual, obtemos o fator de escala correspondente ao tempo em que o sinal observado foi emitido.

2.2 Redshift Espectroscópico

A espectroscopia é uma técnica que estuda a luz decomposta em um espectro eletromagnético, ou seja, em todos os comprimentos de ondas possíveis. Existem três tipos de espectro: como um espectro contínuo, produzido por um corpo opaco e quente; um espectro de emissão, produzido por um gás transparente e quente, que apresenta linhas brilhantes formadas quando há transição do elétron de um estado de maior energia para um estado de menor energia, emitindo um fóton; e um espectro de absorção, que apresenta linhas escuras, formadas quando há a absorção de fótons por um gás relativamente frio interposto entre o observador e a fonte. A descrição desses tipos de espectro é oriunda das Leis de Kirchhoff para a espectroscopia (de Souza Oliveira Filho, 2017).

Cada elemento químico tem energias de transição que resultam em linhas de emissão (ou absorção) características, como uma impressão digital, que permitem identificar a composição química de um gás. Além disso, o espectro permite a determinação de propriedades físicas do objeto estudado, o que faz com que essa técnica seja muito utilizada no estudo de objetos astronômicos.

Diante disso, é possível fazer a medição do redshift cosmológico através da espectroscopia da galáxia, sendo este chamado de redshift espectroscópico (spec-z). Uma vez obtido o espectro da galáxia, os comprimentos de onda das linhas presentes para cada elemento químico são comparadas com as linhas dos espectros medidos na Terra para esses mesmos elementos, da seguinte forma:

$$z = \frac{\lambda_{obs} - \lambda_{ref}}{\lambda_{ref}}, \tag{2.4}$$

onde λ_{obs} é o comprimento de onda no espectro da galáxia e λ_{ref} é o comprimento de onda medido na Terra.

Dessa maneira, a espectroscopia fornece um valor muito preciso de redshift. Porém, é uma técnica mais custosa e apresenta uma limitação observacional para objetos distantes, uma vez que há a dispersão da luz por um prisma (ou rede de difração) para a obtenção do espectro, o que faz com que o sinal/ruído mínimo exigido seja maior do que na fotometria, além de haver restrições em relação à quantidade de objetos que o telescópio é capaz de observar simultaneamente. O levantamento DESI revolucionou o uso da técnica com a aplicação de fibras robóticas, aumentando a quantidade de espectros captados por noite; ainda assim, não supera a técnica de fotometria em relação à cobertura de objetos, com a qual é possível determinar o redshift até mesmo de artefatos. Essas questões impactam principalmente na quantidade de objetos que os levantamentos espectroscópicos retornam, o que se torna um problema em grandes levantamentos cosmológicos (Soo, 2018).

Como alternativa a essas barreiras, começa-se o uso da fotometria para a determinação do valor do redshift, que será explorado na próxima seção.

2.3 Redshift fotométrico

2.3.1 Fotometria para obtenção do redshift

A fotometria é a medida da quantidade do fluxo de fótons em determinado intervalo de comprimento de onda. O fluxo pode ser definido como a quantidade de energia da radiação eletromagnética recebida por unidade de área e tempo (de Souza Oliveira Filho, 2017). A partir da medida do fluxo, podemos determinar a medida da magnitude aparente do objeto de acordo com a equação

$$m = -2.5 \log(F) + C, \tag{2.5}$$

onde F é o fluxo medido do objeto e C é a constante do ponto de referência do sistema de magnitude utilizado. Os dados utilizados neste trabalho usam o sistema de magnitude AB, sendo a unidade adotada para os valores de fluxo o $nanoJanksy(nJ_y)$ [1] e constante $C = 34.1$ (Ivezić, 2019).

As magnitudes são obtidas pela medição do fluxo através de filtros ou bandas passantes. Cada filtro delimita um intervalo de comprimento de onda que é permitido passar pelo mesmo e possui uma curva de transmissividade característica, determinada pelas propriedades ópticas do material que o constitui. Um conjunto bem definido de filtros em diferentes comprimentos de onda determinam um sistema fotométrico.

Os filtros utilizados no sistema fotométrico variam de acordo com o interesse do estudo científico. O LSST utiliza o sistema $ugrizy$, com filtros de banda larga abrangendo seis bandas do espectro eletromagnético, do ultravioleta próximo ao infravermelho próximo $(320 - 1050nm)$ (Ivezić, 2019), como ilustrado na Figura 2.2.

A partir das medidas de magnitudes de um sistema fotométrico, pode-se definir também os índices de cor de um objeto, caracterizados pela diferença entre as magnitudes de duas bandas. Quanto menor a diferença, mais azul é a cor (e quanto maior, mais vermelha) e, a partir dessas cores, podemos obter informações físicas e químicas sobre os objetos estudados.

A partir da fotometria da galáxia - os valores de magnitudes obtidos em cada banda fotométrica - utilizando amostras espectroscópicas como base, pode-se obter o

[1] $1 J_y = 10^{-23} ergs^{-1} Hz^{-1} cm^{-2}$

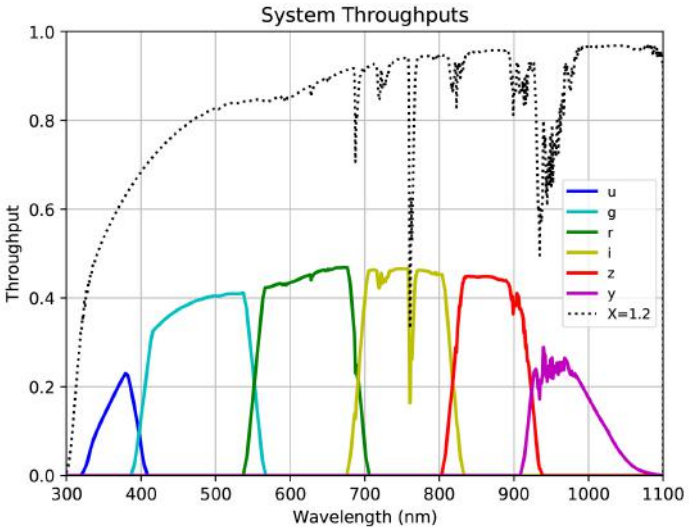


FIGURA 2.2. Bandas passantes do LSST. As linhas coloridas indicam as faixas de comprimento de onda(eixo horizontal) e a transmissão total (eixo vertical) de cada banda. A linha pontilhada mostra a transmissão atmosférica, para uma massa de ar de 1.2 (Ivezić, 2019).

valor do redshift, tendo assim o que se chama de redshift fotométrico (photo-z). A medição do redshift por este método apresenta uma perda na precisão das medidas individuais; porém, a fotometria possibilita a obtenção de um grande volume de dados, o que compensa estatisticamente a diminuição dessa precisão.

Os métodos para medição de redshifts fotométricos podem ser divididos em métodos de ajuste de modelos e métodos empíricos, que serão discutidos a seguir.

2.3.2 Métodos de estimativa de z_{phot}

Os métodos para calcular photo-zs podem ser classificados em dois grandes grupos: métodos de ajuste de modelos e métodos empíricos. O método de ajuste de modelos é feito através do ajuste de modelos baseados em espectros prévios, chamados de *Spectral Energy Distributions* (SED), para estimar o redshift da galáxia. Um conjunto pré-definido de espectros em diferentes redshifts são ajustados com a fotometria obtida para a galáxia e, então, o fluxo observado é comparado com os fluxos do modelo teórico, fazendo-se um teste de χ^2-quadrado para encontrar o redshift que melhor se ajusta ao observado, correspondente ao modelo que apresentar o menor valor de χ^2-quadrado (Soo, 2018). Na Figura 2.3, temos um exemplo de ajuste de dados observacionais a fluxos baseadas nos modelos.

Como vantagens desse método podemos citar que ele permite a caracterização dos objetos com a associação a um tipo espectral, que pode ser relacionado a tipos morfológicos, ou outras classificações; também pode-se utilizar modelos sintéticos baseados em síntese de populações estelares e isso permite associar resultados à física dos objetos (estimar massa bariônica, idade, metalicidade, taxa de formação estelar etc); e não dependem da disponibilidade de um *training set* representativo e por isso também podem ser utilizados de forma complementar para estimar photo-zs em regiões do hiperespaço de cores cuja cobertura espectroscópica é insuficiente. Em contrapartida, podem ser mais demorados e computacionalmente custosos, além de serem muito sensíveis às degenerescências na medição do photo-z. Também são menos flexíveis na adição de novos atributos de entrada que possam melhorar os resultados.

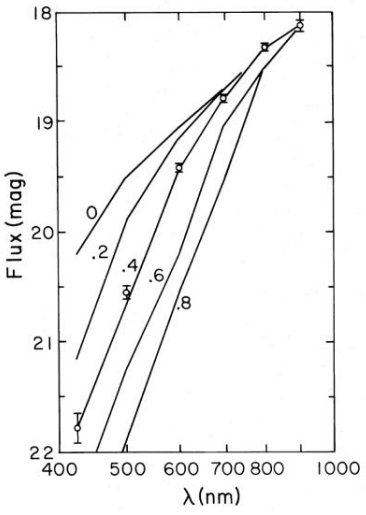


FIGURA 2.3. Exemplo de estimativa de redshift de uma galáxia pelo método de template. As barras de erros indicam os dados observacionais da galáxia e as linhas de referência são os modelos em diferentes faixas de redshift. Observamos que os dados se ajustam melhor à linha na faixa de $z \sim 0.4$, sendo o valor estimado $z = 0.398 \pm 0.018$ (Loh & Spillar, 1986).

Os métodos empíricos se baseiam na construção de equações empíricas que relacionam a fotometria do objeto com o valor de redshift, determinadas a partir de dados observacionais de galáxias com redshifts espectroscópicos conhecidos e, a partir dessas relações, são determinados os redshifts fotométricos.

Com o avanço da tecnologia, os métodos empíricos passaram a ser ligados ao uso de machine learning para determinação de redshifts fotométricos. A técnica de machine learning consiste em um algoritmo que recebe um conjunto de medições com redshifts espectroscópicos conhecidos, aprende as relações entre os parâmetros e, assim, é capaz de medir redshifts fotométricos em outros conjuntos com redshifts indeterminados.

Os métodos de machine learning apresentam redshifts fotométricos com menor dispersão em relação aos redshifts espectroscópicos em geral, principalmente em valores intermediários de redshifts, por terem maior quantidade de conjuntos para treinamento (Soo, 2018). Este é o método utilizado neste trabalho e será detalhado na próxima seção.

2.4 Machine Learning

Machine Learning é uma área da Inteligência Artificial baseada na capacidade das máquinas em reconhecer padrões para tomada de decisões. O algoritmo recebe um conjunto de dados com valores conhecidos das grandezas ou características que se deseja medir e constrói, sem um conjunto de regras prévio, a relação entre eles, de modo que as decisões tomadas se adaptam para que o resultado final seja o mais acurado possível (Marsland, 2015). Desta forma, podemos utilizar a relação criada para fazer a predição das mesmas grandezas para conjuntos onde estas são desconhecidas. A Figura 2.4 apresenta um diagrama de comparação entre as estruturas da programação clássica e do ML.

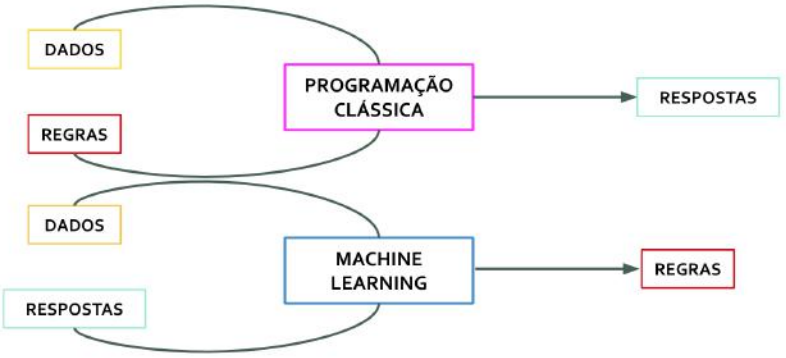


FIGURA 2.4. Comparação entre as estruturas da programação clássica e do ML. Adaptado de Chollet (2017).

Esse método se mostra muito útil na medida de redshifts fotométricos por possibilitar a construção de modelos analíticos complexos, uma vez que não há uma relação matemática linear entre as medidas de magnitudes e os valores de redshifts.

Existem diversos algoritmos já utilizados para determinação de photo-zs, dentre eles Artificial Neural Networks (ANNs) (Collister & Lahav, 2004), FlexZBoost (Izbicki & Lee, 2017), GPz (Almosallam et al., 2016) e METAPhoR (Cavuoti et al., 2017).

2.4.1 Classificação e Regressão

Entre as diferentes formas de se aplicar o machine learning, os dois tipos mais comuns de algoritmos são os de aprendizados supervisionados e os não-supervisionados.

No aprendizado supervisionado, é fornecido ao algoritmo um conjunto de dados com valores conhecidos da grandeza que se deseja inferir e, então, a máquina busca uma relação entre os dados e o valor final para que o modelo seja posteriormente aplicado em dados sobre os quais não se conhece o valor (por exemplo, o redshift). No aprendizado não supervisionado, os dados são fornecidos sem as respostas e o algoritmo precisa aprender sozinho as relações dos dados com os resultados que se busca.

O aprendizado supervisionado possui duas técnicas de aplicação: classificação e regressão.

Seja um conjunto de dados com os pares (x, y), sendo x o valor de entrada (*features*) e y seu valor de saída (*targets*), sobre os quais se deseja fazer a previsão. A classificação tem como objetivo construir um modelo que categorize os dados em classes específicas, dentro de um conjunto finito de possibilidades, representadas por y no conjunto definido acima. A partir das classificações dadas no conjunto de entrada, o algoritmo busca encontrar as condições de contorno que farão com que o modelo consiga distinguir quais valores de x pertencem a uma mesma classe e agrupá-los (Andrew Ng, 2022). A classificação pode ser feita pela abordagem generativa, na qual é feita uma modelagem de como os dados se distribuem em cada classe, ou pela abordagem discriminativa, que foca em encontrar as fronteiras de separação das classes diretamente, sendo útil em casos onde há alta dimensionalidade dos atributos (Zěljko Ivezić & Gray, 2020).

Para avaliar se a classificação está funcionando bem, é usada uma função que avalia o desempenho do modelo. Essa função é chamada de *função de custo* e, para a classificação, mede o quanto o modelo é capaz de acertar a classe na qual ele coloca o dado. A função de custo mais comum na classificação é a Perda zero-um, definida por

$$P(y_{real}, y_{pred}) = \begin{cases} 1, & \text{se } y_{real} \neq y_{pred} \\ 0, & \text{se } y_{real} = y_{pred} \end{cases}, \tag{2.6}$$

e é interpretado de modo que seu valor é 0 quando modelo acerta a previsão da classe e 1 quando erra. A partir disso, é possível calcular o risco de classificação, definido por

$$E[L(y_{real}, y_{pred})] = P(y_{pred} \neq y_{real}), \tag{2.7}$$

que quantifica a probabilidade do modelo errar a classe (Zěljko Ivezić & Gray, 2020).

Já na regressão, o modelo visa associar os dados a um valor numérico dentro de um conjunto infinito de possibilidades, limitado pela discretização imposta pela precisão numérica, que seriam também representados por y no conjunto definido. Supondo que se queira encontrar um valor de y para um x não pertencente ao conjunto de dados, o problema de regressão busca, então, uma função que relacione os pares (x, y) dados, ajustando os valores de x em uma curva que abranja o máximo de pontos possível, para que seja possível prever o valor de y para qualquer x (Marsland, 2015). Podemos expressar a função como

$$y = f(x|\boldsymbol{\theta}), \tag{2.8}$$

sendo $\boldsymbol{\theta}$ os parâmetros a serem definidos para o modelo. Para um modelo específico, com k parâmetros: θ_1, θ_2, ..., θ_k , cada observação (x_i, y_i) impõe uma restrição no espaço de parâmetros e temos as regiões de probabilidade para os valores dos mesmos, associadas às incertezas dos dados de entrada (Zěljko Ivezić & Gray, 2020). O desempenho do modelo está relacionado à escolha dos melhores parâmetros para a função que relaciona x e y. Os melhores parâmetros serão os que minimizam a função de custo, que na regressão mede o quanto os valores preditos se distanciam do valor real (Andrew Ng, 2022). A função de custo mais comum é o MSE (traduzindo do inglês, Erro Quadrático Médio), definido como a média do quadrado das diferenças entre os valores preditos e os valores reais, dada por

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_{pred,i} - y_{real,i})^2. \tag{2.9}$$

O MSE penaliza mais erros maiores, reduzindo grandes discrepâncias. Outras funções de custo podem ser utilizadas, dependendo da estimativa a ser feita, como: MAE (traduzindo do inglês, Erro Absoluto Médio), que é menos sensível a outliers; RMSE (traduzindo do inglês, Raiz do Erro Quadrático Médio), que é a raiz quadrada do MSE, tendo as mesmas unidades da variável de saída, o que facilita a interpretação dos erros; CDE loss (traduzindo do inglês, Perda por Estimativa de Densidade Condicional), usado quando o modelo retorna uma distribuição de probabilidade completa para cada predição, entre outros.

O tipo de regressão utilizada depende de alguns fatores, que podem ser elencados em três aspectos principais: **(i)** a **linearidade**, pois a relação entre os parâmetros e o modelo pode ser linear, ou seja, pode ser escrito como $f(x|\theta) = \sum_{p=1}^{k} \theta_p g_p(x)$, ou não linear, como em $f(x|\theta) = \theta_1 + \theta_2(sin\theta_3 x)$; **(ii)** a **complexidade do problema**, que pode ser medida pela dimensão de x, ou seja, quanto mais variáveis independentes, maior será a complexidade; **(iii)** o **comportamento dos erros** das variáveis dependentes e independentes, já que as incertezas nas medições tem influência direta no desempenho do modelo (Zěljko Ivezić & Gray, 2020).

2.4.2 Estrutura de um algoritmo de Machine Learning supervisionado

Embora existam diversos algoritmos de machine learning supervisionados, todos seguem o mesmo conceito. Vamos utilizar aqui como exemplo ilustrativo o objetivo deste trabalho, a obtenção do redshift fotométrico de um conjunto de galáxias.

Primeiramente, é utilizado no algoritmo um *training set*, que consiste em um conjunto de galáxias com suas magnitudes em cada banda (*features*) e os valores de redshifts espectroscópicos já conhecidos (*targets*). O algoritmo, então, deve construir as relações entre as magnitudes em cada filtro e seus redshifts e armazenar essas equações. Os parâmetros de ajuste da equação são atualizados à medida que o algoritmo faz o treinamento, e este acaba quando há uma distância mínima entre o valor real e o valor gerado.

Um ponto crítico que pode ocorrer ao se aplicar um *training set* é o *overfitting*; quando os resultados do modelo gerado pelo algoritmo são muito parecidos com esse conjunto, isso reflete que o algoritmo construiu relações apenas para aquele conjunto

específico, o que resulta na baixa performance nas medições para conjuntos diferentes. Como solução, pode ser feita a etapa de *cross-validation*, onde é utilizado um conjunto de validação, que será um conjunto diferente do anterior, mas ainda com redshifts conhecidos. Comparando os resultados de redshifts do conjunto de treino com os do conjunto de validação para diferentes níveis de complexidade do modelo utilizado, determinamos como o nível de complexidade ideal aquele que apresenta o menor erro para os dois conjuntos (Zěljko Ivezić & Gray, 2020). A complexidade será definida pela escolha de hiperparâmetros, que são os parâmetros de configuração do algoritmo a serem definidos pelo usuário.

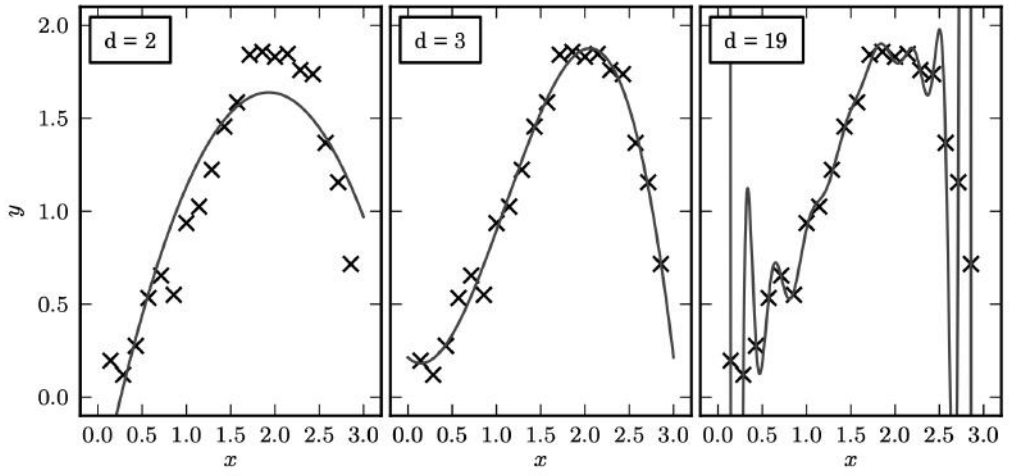


FIGURA 2.5. Complexidade ideal do modelo. O ajuste à esquerda representa um modelo pouco complexo, causando um viés alto. O ajuste à direita mostra um modelo muito complexo, causando um alta variância. O ajuste no centro mostra a complexidade ideal, com um bom ajuste (Zěljko Ivezić & Gray, 2020).

Estabelecidas e validadas as relações entre redshifts e magnitudes, utilizamos um conjunto diferente de objetos com propriedades similares aos conjuntos anteriores, chamado conjunto de teste para avaliar o modelo gerado pelo algoritmo. Esse conjunto possui spec-zs conhecidos, mas apenas as magnitudes são utilizadas nesta etapa. Os photo-zs produzidos para esse conjunto são usados para avaliar a performance do algoritmo através de métricas escolhidas, em comparação com os spec-zs, sendo úteis também para a comparação entre diferentes algoritmos. Os resultados obtidos nas métricas quantificam as incertezas das predições, informação essencial para análises futuras que utilizem o photo-z. Sendo a performance avaliada , vamos para a etapa do conjunto alvo, com galáxias sem redshifts conhecidos, que é o conjunto que queremos medir.

Dentre as diferentes abordagens oferecidas pelos algoritmos para executar as etapas descritas acima, escolhemos utilizar neste trabalho o método de árvores de decisão

aplicadas às florestas aleatórias, que será explorado na próxima seção.

2.4.3 Árvores de decisão e Florestas Aleatórias

A árvore de decisão é um algoritmo que se baseia na tomada de decisão hierarquizada, dividindo o conjunto de dados inicial em subconjuntos de acordo com o critério de decisão estabelecido. A árvore se inicia com nó raiz, que apresenta a decisão de contorno inicial, e então o conjunto de dados é dividido em dois novos subconjuntos: um com os dados que estão dentro da decisão de contorno e outro com os dados que estão fora da mesma, como ilustrado na Figura2.6. Para cada conjunto, temos dois novos nós de decisão e o algoritmo segue assim até que se chegue ao valor final, de acordo com o critério de parada (Zěljko Ivezić & Gray, 2020).

Como essas decisões são definidas depende do tipo de árvore de decisão e do algoritmo. As árvores podem ser aplicadas tanto em problemas de classificação quanto de regressão.

Na árvore de classificação, o critério de decisão é definido de acordo com o atributo que traz a informação mais relevante sobre as classes. Isso é quantificado a partir do ganho de informação, definido como

$$I_{ganho}(D, A) = I_{impureza}(D) - \sum_{m \in \, valores \; de \; A} \frac{|D_m|}{|D|} I_{impureza}(D_m), \tag{2.10}$$

onde D são os dados no nó atual, A é o atributo que está sendo testado como divisor do nó, m são os possíveis valores se A, D_m são os dados no subconjunto e $I_{impureza}$ é o índice de grau de impureza. Este último pode ser calculado de três formas, a depender do algoritmo. Seja n o número de classes possíveis e p_i o subconjunto de dados de D pertencentes à classe i. Os índices de grau de impureza podem ser:

- Entropia, que quantifica o grau de desordem dos dados; quanto menor a entropia, maior o ganho de informação:

$$I_{impureza} = -\sum_{i=1}^{n} p_i \; log_2 \; p_i \; ; \tag{2.11}$$

- Coeficiente de Gini, que considera a pureza do nó a partir da quantidade de dados de uma mesma classe:

$$I_{impureza} = 1 - \sum_{i=1}^{n} p_i^2 \ ; \tag{2.12}$$

- Erro de classificação, que mede a quantidade de dados que estão numa classe diferente da marjoritária do nó:

$$I_{impureza} = 1 - max(p_i). \tag{2.13}$$

As árvores de regressão seguem o mesmo conceito de divisão recursiva das árvores de classificação, porém, no lugar de divisão e classes, são feitas predições com modelos de regressão para os dados de cada folha, retornando a média dos valores de redshift preditos. O critério de divisão também é escolhido de forma diferente; nesse caso, são usados o MSE (mais comum) ou o MAE (usado em alguns casos), para medir a dispersão dos valores preditos no nó. O atributo que minimizar essa dispersão será o escolhido (Carrasco Kind & Brunner, 2013).

Utilizar apenas uma árvore de decisão para fazer a predição pode não ser um bom caminho, uma vez que, como a determinação do nó raiz está atrelada a incertezas, podemos não estar trabalhando de fato com a melhor árvore. Para obter um resultado mais robusto, utilizamos o algoritmo de floresta aleatória.

A floresta aleatória é um conjunto de árvores de decisão criadas a partir de subconjuntos da amostra original. Esse algoritmo se baseia em uma técnica denominada *bagging*, onde temos o uso de múltiplos modelos (árvores) combinando suas previsões para o resultado final. Os conjuntos utilizados em cada árvore são criados através da técnica de reamostragem *bootstrap*, na qual temos k conjuntos de treino criados aleatoriamente a partir do conjunto de dados original, de mesmo tamanho N da amostra original, através de sorteio com reposição (Zěljko Ivezić & Gray, 2020). Uma vez criados os conjuntos e, em seguida, as respectivas árvores de decisão, a predição final será a média das predições de cada árvore. A Figura 2.7 ilustra esse processo.

Dada a fundamentação teórica apresentada, dirigimos, no próximo capítulo, o foco à metodologia aplicada para a estimativa dos photo-zs.

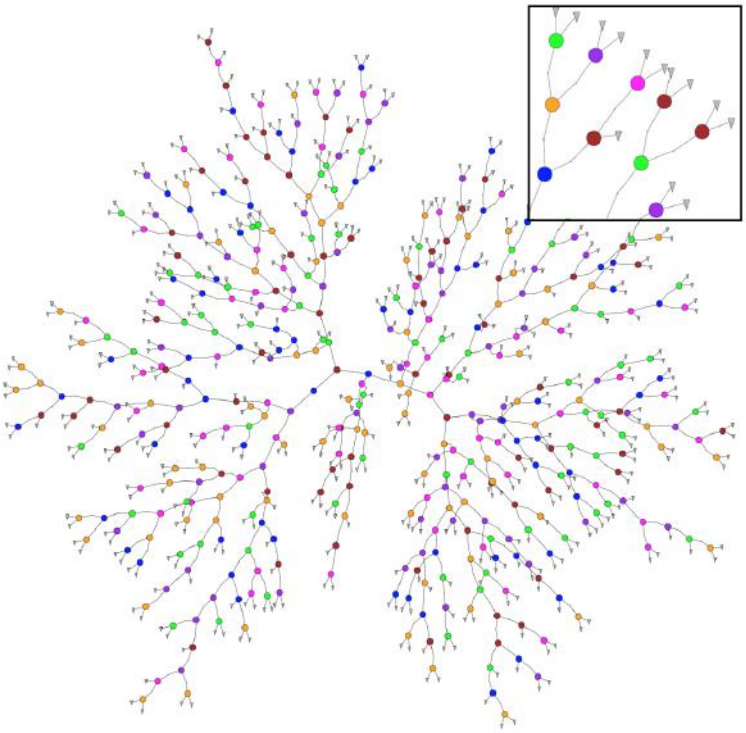


FIGURA 2.6. Exemplo de árvore de decisão. Cada cor representa uma decisão de contorno. Figura retirada de Carrasco Kind & Brunner (2013)

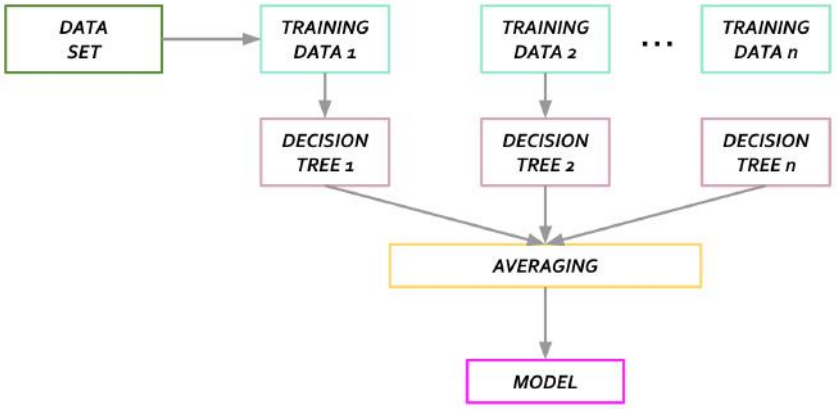


FIGURA 2.7. Ilustração do processo de criação de uma floresta aleatória de árvores de decisão.

Capítulo 3

Estimativa de photo-z nos dados simulados

Neste capítulo, serão descritas as ferramentas computacionais utilizadas e as etapas para estimar os photo-zs. A geração do resultado final global passa por duas etapas: primeiro, há a geração do modelo com uma pequena fração do conjunto de dados, usada para treinar e validar o algoritmo. Esta etapa foi feita em Jupyter Notebooks no ambiente Jupyter Lab, detalhados no Apêndice B, uma vez que esta etapa não envolve big data e não se faz necessário o uso do cluster; a segunda etapa foi realizada no cluster de computadores Apollo, a partir da aplicação do modelo gerado na etapa anterior ao conjunto de dados completo, sem o valor de redshift conhecido, para geração do catálogo final.

3.1 Algoritmo: TPZ

O TPZ – *Trees for Photo-Z* – é um algoritmo de aprendizado supervisionado, escrito na linguagem de programação Python e com processamento em paralelo usando MPI, que utiliza o método de árvores de decisão, por meio das florestas aleatórias, para estimar redshifts fotométricos (Carrasco Kind & Brunner, 2013). Dadas as informações fotométricas e os spec-zs das galáxias do conjunto de dados, o TPZ retorna as funções de densidade de probabilidade (do inglês, PDF) dos photo-zs. O fato de termos uma PDF como resultado para cada galáxia é uma abordagem muito relevante neste problema,

uma vez que a relação entre magnitudes e redshift pode ser degenerada, ou seja, objetos com magnitudes semelhantes podem ter diferentes valores de redshift. Assim, essas incertezas na predição são capturadas ao retornar uma densidade de probabilidade ao invés de um único valor de redshift.

Os modos de classificação e regressão foram implementados originalmente no algoritmo TPZ. No modo de classificação, a amostra é dividida em bins de redshift e para cada um deles é criada uma floresta aleatória. É feita uma seleção de quais galáxias serão treinadas em cada bin, de modo a não comprometer a performance do algoritmo e diminuir a chance de erros catastróficos. Feito isto, o conjunto de dados é treinado em cada floresta, sendo feita a classificação de cada galáxia como dentro ou fora da faixa de redshift. Combinando todos os resultados, é dada a probabilidade de cada galáxia estar na faixa de redshift e é gerada a PDF. No modo de regressão, não há divisão em bins; o conjunto completo é utilizado para criar os subconjuntos e treinar a floresta em toda faixa de redshift espectroscópico. A árvore é iniciada com um nó contendo todos os dados e é dividida recursivamente de acordo com a dimensão com o maior ganho de informação, sendo o critério de divisão feito através da minimização da soma dos erros quadrados.

A versão original do TPZ foi escrita em Python 2, e há uma atualização escrita em Python 3, que foi utilizada neste trabalho, que chamaremos de TPZ Lite. Essa versão está implementada no RAIL (Team et al., 2025), projeto *open-source* desenvolvido pelo LSST-DESC para estimativa e análise de redshifts fotométricos, que será descrito com mais detalhes na seção 3.2.

Para a construção da floresta aleatória, são criados n catálogos a partir do catálogo original por meio de uma perturbação gaussiana nos valores das features, a partir dos erros dessas medidas. Para cada catálogo, são criadas m amostras *bootstrap* para fazer a floresta aleatória. Os valores de n e m são definidos pelo usuário, sendo o n x m o número total de árvores da floresta (Carrasco Kind & Brunner, 2014), como ilustrado na Figura 3.1. Os valores de n e m são definidos pelos hiperparâmetros *nrandom* e *ntree*, respectivamente.

Como parte da adaptação para Python 3, no TPZ Lite é possível escolher o modo como as árvores serão construídas, através do hiperparâmetro *tree strategy*. Escolhendo a estratégia *native*, as árvores são construídas de acordo com o código original TPZ.

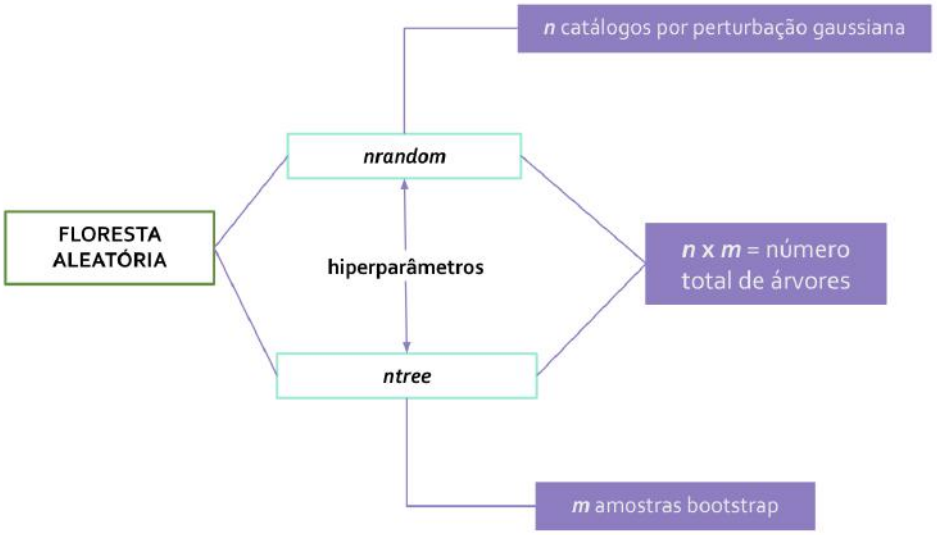


FIGURA 3.1. Ilustração do processo de criação da floresta aleatória no algoritmo TPZ.

Porém, esse método se mostra muito custoso computacionalmente e em tempo; por isso, foi implementada a opção *sklearn*, na qual as árvores são construídas utilizando o *DecisionTreeRegressor* da biblioteca *Scikitlearn* (Pedregosa et al., 2011). O uso dessa estratégia pode reduzir o tempo de treinamento em até 10.000 vezes e possui resultados qualitativamente similares ao TPZ original, sendo a escolhida para este trabalho (Schmidt, 2025).

O critério de divisão em cada nó é feito com base no grau de impureza, sendo escolhida a divisão que minimiza a impureza, como visto na seção 2.4.3. Seja um nó m, com n dados. Um determinado critério de divisão baseado em uma feature vai dividir os n dados em dois nós filhos, com $n_{esquerdo}$ e $n_{direito}$ dados. A melhor divisão será avaliada por uma média ponderada da função de impureza de cada nó filho. Para a regressão, a função de impureza é dada pelo erro quadrado médio (MSE), de modo que

$$MSE = \frac{1}{n}\sum_{i=1}(y_i - y_m)^2, \tag{3.1}$$

sendo y_m a média dos valores naquele nó. Assim, a média ponderada para definir o critério de divisão é dada por

$$M = \frac{n_{esquerdo}}{n} \cdot MSE_{esquerdo} + \frac{n_{direito}}{n} \cdot MSE_{direito}, \tag{3.2}$$

sendo escolhido o critério com menor valor de M (Pedregosa et al., 2011).

O hiperparâmetro *n_att*, definido pelo usuário, determina um subconjunto de $n_* < N$ - sendo N o número total de atributos - escolhidos aleatóriamente em cada nó para escolha do critério de divisão. Isso evita que as árvores sempre usem atributos semelhantes e consigam capturar diferentes padrões nos dados (Carrasco Kind & Brunner, 2013). Uma vez que o critério é escolhido, então os dados são finalmente divididos em dois novos nós e esse processo é, então, repetido para cada novo nó, até que se atinja um critério de parada, definido pelo hiperparâmetro *minleaf*. Esse parâmetro determina o número mínimo de dados para que a árvore continue a fazer a divisão em um nó; não sendo atendido o número estabelecido, chegamos no critério de parada da recursão.

Ao fim da divisão em todas as árvores, temos um pequeno grupo de galáxias nas folhas terminais. É feita uma média simples dos valores de redshift de cada folha, retornando esse valor de redshift para cada galáxia daquela folha. Então, é construído um histograma com os valores de redshift atribuídos para cada galáxia em todas as árvores da floresta, e, assim, é construída sua PDF (Schmidt, 2025).

3.2 RAIL

O RAIL é uma biblioteca de software open-source escrita em Python que tem como objetivo prover ferramentas para geração de redshifts fotométricos em escala, fornecendo também incertezas e estatísticas associadas e permitindo testes em cenários realísticos. Seu desenvolvimento foi motivado a partir das experiências com o Data Challenge 1 (DC1), uma simulação de dados esperados para o LSST, visando estruturar as etapas de estimação dos resultados do Photometric Redshifts Working Group, seguindo as pipelines do DESC, e sua infraestrutura é útil em diversas aplicações científicas do LSST (Team et al., 2025).

O RAIL possui três subpacotes: o *creation*, que possui módulos com ferramentas para criação de training sets com ferramentas que possibilitam desde a criação de training sets realisticamente complexos, a partir de dados simulados; o *estimation*, com o qual é realizada a execução do algortimo para gerar os photo-zs; e o *evaluation*, onde temos diferentes métricas para avaliação da performance do algoritmo, permitindo um processo de ponta-a-ponta na geração de catálagos de photo-zs.

No subpacote *estimation*, temos os métodos *inform*, no qual é feito o treino para construção do modelo, e o *estimate*, onde o modelo criado no *inform* é executado no conjunto de dados para fazer a predição dos valores de redshift. Eles são importados de acordo com o algoritmo a ser usado; por exemplo, para o TPZ temos o TPZInformer para o método *inform* e o TPZEstimator para o método *estimate*. Na Figura 3.2, temos a ilustração da estrutura dos subpacotes descritos.

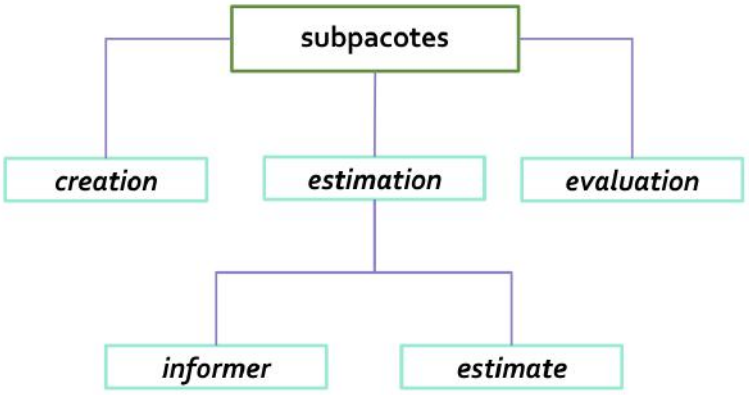


FIGURA 3.2. Ilustração da estrutura de subpacotes do RAIL.

Como saída do método *estimate*, temos um objeto do tipo *qp.Ensemble*, através do pacote qp (*quantile parametrization*), que armazena as PDFs de forma flexível e adaptável para diferentes análises científicas. Ele cria uma interface comum para as diferentes parametrizações das PDFs, fornecendo ferramentas para as diferentes formas de representação e cálculo de métricas. Seus principais recursos são: *metadata*, que contém o tipo de parametrização; *objdata*, contendo os parâmetros de cada PDF, de acordo com *metadata*; e *ancil*, com informações auxiliares que não fazem parte da parametrização da PDF. Essas informações podem ainda ser salvas em um arquivo de saída para análises posteriores.

3.3 Pz Compute

Como parte da contribuição *in-kind* para o LSST, na qual são fornecidos apoios não financeiros em troca de direito de acesso aos dados, o LIneA fornecerá tabelas de photo-zs como um conjunto de dados federados a cada novo *data release*, com um método de estimativa diferente do utilizado nas estimativas produzidas pela equipe de gestão de

dados (DM), que serão disponibilizadas no catálogo fotométrico anual, com o objetivo de expandir o escopo científico abrangido pelos dados divulgados. Essas tabelas serão produzidas e armazenadas na estrutura do IDAC-Brasil. Os IDACs são centros de dados fora dos EUA que possuem autorização para comportar e distribuir os dados do LSST.

Para esta tarefa, foi desenvolvido pelo LIneA o pz-compute, um pipeline para calcular photo-zs usando códigos públicos para grandes volumes de dados, construído com base no RAIL (LIneA (Laboratório Interinstitucional de e-Astronomia), 2025). Os métodos de estimativa de photo-z são aplicados ao conjunto de dados utilizando a infraestrutura do cluster de computadores Apollo, detalhada no Apêndice C.

O `pz-compute` possibilita a customização de parâmetros de configuração do Slurm[1], permitindo otimizações de desempenho baseadas no balanceamento de carga, considerando características do *cluster*, como número de núcleos de processamento, memória disponível em cada nó e o particionamento dos dados. Além disso, o `pz-compute` possui ferramentas de gestão de processos que registram metadados necessários à reprodutibilidade dos resultados.

A geração das tabelas de photo-zs com o pz-compute passa por uma série de etapas, sendo elas:

- Aquisição de dados: algumas semanas antes de cada DR, os dados são transferidos para o IDAC-Brasil, registrados e validados; são alocados na área de armazenamento de alta velocidade (Lustre FS), visível pelos nós do cluster;
- Pré-processamento: a seguir, é feita a produção das *skinny tables*, que são tabelas reduzidas contendo apenas as colunas de interesse para gerar as tabelas de photo-z, além da aplicação de tratamentos de interesse como correção de avermelhamento, conversão de fluxos para magnitudes, arredondamento de casas decimais extras, etc;
- Pré-processamento: aplicação do pipeline *Training Set Maker*, com o qual é feito o cross-matching espacial para a associação dos dados fotométricos com os spec-zs previamente obtidos de outros levantamentos ou redshifts verdadeiros, em caso de dados simulados, reproduzindo o procedimento completo como preparação para o futuro com dados reais. Após a caracterização do conjunto, onde é feita a

[1]Sistema de gerenciamento de cargas de trabalho do *cluster* Apollo

descrição e análise das propriedades observacionais e de cobertura do mesmo, são executados o *inform*, o *estimate* e o *evaluation* para gerar o relatório de métricas de desempenho do algoritmo. O conjunto de treinamento, o modelo gerado e os resultados da validação do photo-z são salvos na plataforma PZ Server - também desenvolvida pelo LIneA dentro da contribuição *in-kind* -, onde serão hospedados os produtos de dados leves, permitindo o compartilhamento entre os usuários com direito aos dados;

- PZ Compute: sendo aprovado o relatório de métricas, o pipeline pz-compute é, então, executado em todo conjunto de dados para gerar a tabela final e os metadados são armazenados no PZ-Server

- Pós-processamento: entrega das tabelas armazenadas no IDAC-Brasil para o USDAC e registro das mesmas na *Rubin Science Platform* (RSP), como uma tabela de dados federados, disponibilizando para os usuários. A transferência de dados em ambas as direções é um desafio importante, e parte da contribuição in-kind do Brasil consiste na instalação da rede de fibras ópticas que, vinda do Chile, seguirá para os Estados Unidos passando por território brasileiro.

A Figura 3.3 traz uma ilustração do fluxograma do fluxo de dados no IDAC-Brasil, com destaque para o fluxo da geração das tabelas de photo-zs. A partir da transferência vinda dos EUA, chegando na *Data Transfer Node* (DTN), os dados são alocados no sistema de armazenamento que alimenta o ambiente HPC. Em seguida, são pré-processados, associados via cross-matching espacial a dados de referência (spec-z) para criação de conjuntos de treino e teste, para então gerar estimativas. Esses resultados serão armazenados em banco de dados para alimentar as plataformas científicas, mantidos em backup de longo prazo, assegurando a disponibilidade futura, e enviados novamente para os EUA.

A partir da estrutura do pz-compute, dividimos a estimativa dos photo-zs do catálogo em duas etapas principais: a primeira dedicada à geração e avaliação do desempenho do modelo, e a segunda à aplicação do modelo selecionado à tabela completa, como está ilustrado na Figura 3.4 e será detalhado a seguir.

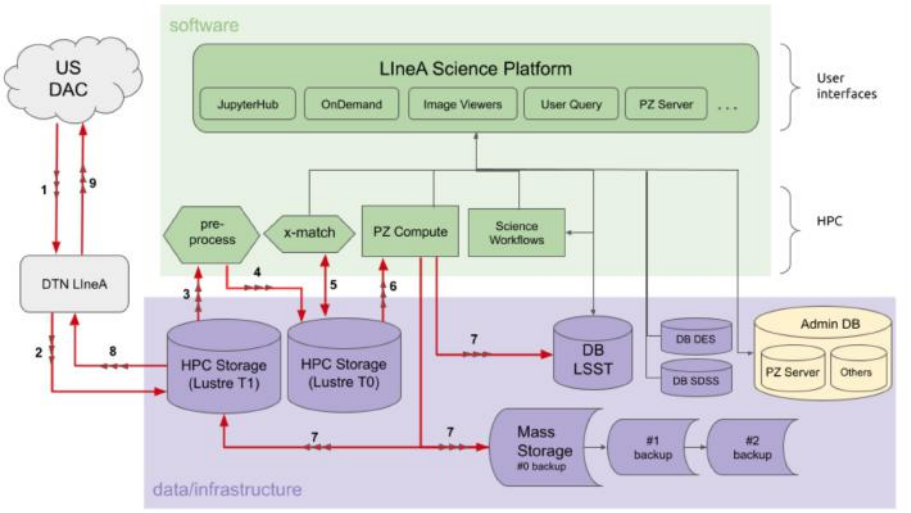


FIGURA 3.3. Fluxo de dados dentro da infraestrutura brasileira do IDAC. As linhas realçadas de vermelho mostram o fluxo de dados relacionado com a produção de photo-zs.

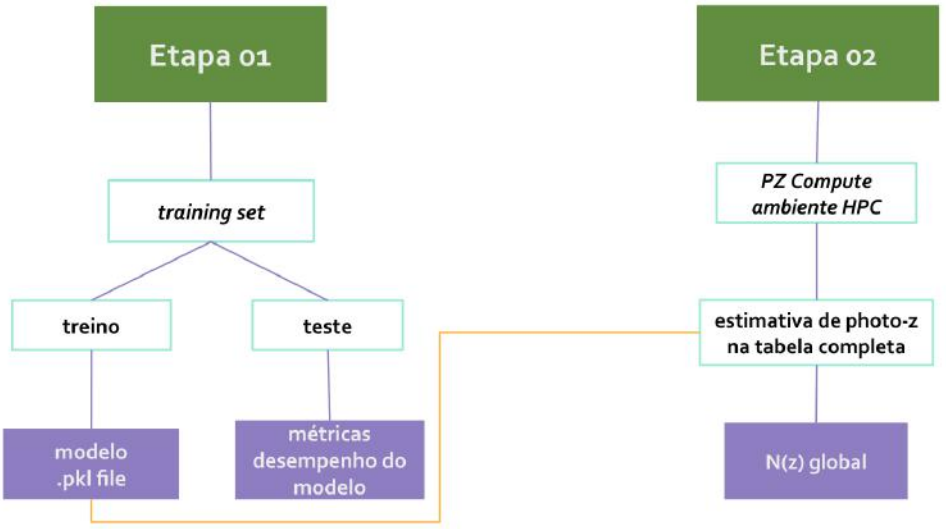


FIGURA 3.4. Ilustração das etapas de estimativa do photo-z.

3.4 Etapa 01: treino, teste e análise das métricas

Como primeiro passo, definimos o *training set* seguindo toda a etapa de pré-processamento do pz-compute. Com o training set definido, fizemos a divisão do mesmo em dois conjuntos: um para o treino e outro para o teste. Essa divisão foi feita baseada nos resultados descritos em Carrasco Kind & Brunner (2013), que indicam melhores resultados dividindo a amostra em 70% dos objetos no conjunto de treino e 30% no conjunto de teste.

Executamos, então, o módulo *inform* com o conjunto de treino e o módulo *estimate* com o conjunto de teste. Os resultados obtidos na execução do *estimate* foram utilizados para calcular as métricas e a analisar o desempenho do modelo, e decidir qual seria utilizado para geração da tabela de photo-z final.

Decidido o modelo a ser utilizado, passamos para a próxima etapa. Todos os passos da etapa 01 foram executados em Jupyter Notebooks e scripts, que estão descritos no Apêndice B.

3.5 Etapa 02: gerando o catálogo de photo-z nos dados simulados

Tendo validado o modelo gerado no treinamento com a análise das métricas, a próxima etapa é aplicar o modelo no conjunto de dados completo, sem valores de redshift conhecidos, para gerar o catálogo final de photo-zs. Isto é feito utilizando o arquivo gerado no treinamento, de formato pickle (.pkl), no módulo *estimate*. Porém, como agora trata-se de um conjunto de dados muito maior, essa aplicação é feita no cluster de computadores através do pipeline pz-compute.

Como etapas de execução, primeiro é feita a ativação do ambiente pz-compute no ambiente HPC do LIneA, via linha de comando. São criados o diretório e subdiretórios de interesse (ex: input, output), e os arquivos de configuração dos parâmetros do cluster e do algoritmo. Colocamos os arquivos da *skinny table* dentro do diretório de *input*, trazemos para o diretório o arquivo .pkl com o modelo a ser utilizado e definimos os parâmetros de configuração. Por fim, executamos o pz-compute na linha de comando para iniciar a estimativa dos photo-zs. Os resultados da execução são armazenados no diretório de *output*.

Após a geração dos photo-zs no catálogo completo, construímos a distribuição global de galáxias por redshifts, e também analisamos a performance da execução do algoritmo no cluster.

As etapas de execução do pz-compute no cluster e a geração do N(z) global estão descritas no Apêndice C.

Capítulo 4

Dados

Como citado no capítulo anterior, o uso de ML está relacionado com a capacidade do computador de reconhecer padrões. Porém, para que os padrões capturados sejam consistentes com a ciência que se quer produzir, é necessário que os dados sejam tratados de forma refinada, pois a performance do algoritmo está fortemente relacionada com a qualidade dos dados passados na entrada. Neste capítulo detalharemos a origem, obtenção e tratamento dos dados utilizados neste trabalho.

4.1 LSST Data Preview 0.2 (DP0.2)

Os dados utilizados neste trabalho foram retirados dos catálogos do LSST Data Preview 0.2 (DP0.2), que consistem em dados simulados construídos a partir das imagens do DESC Data Challenge 2 (DC2), reprocessadas com a versão 23.0.0 do LSST Science Pipelines.

A simulação DC2 foi desenvolvida como parte de um desafio de dados, com o objetivo de criar uma abordagem ponta a ponta para simular os dados esperados para o LSST. Ela fornece um conjunto de dados simulado para o desenvolvimento de pipelines científicos e para compreender as demandas de tamanho e complexidade dos dados futuros. Partindo de uma simulação de N-corpos com levantamentos semelhantes até o processamento para gerar as imagens simuladas com o LSST Science Pipelines, que incluiu diversos efeitos observacionais de acordo com as condições do telescópio, a simulação abrange as seis bandas $ugrizY$ numa área *wide-fast-deep* (WFD) de 300

graus^2 e um *deep drilling field* (DDP) de 1 *grau*^{2}, simulando 5 anos de levantamento (sendo o planejado para o LSST 10 anos) (LSST Dark Energy Science Collaboration (LSST DESC) et al., 2021). Na figura 4.1, há uma ilustração da área de cobertura da simulação.

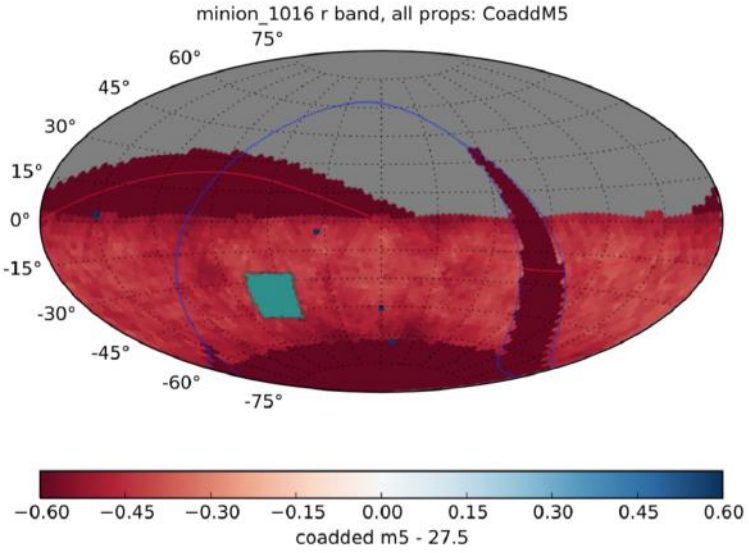


FIGURA 4.1. Área de cobertura da simulação DC2 (região verde) em contraste com a área de corbertura planejada pelo LSST em 10 anos (LSST Dark Energy Science Collaboration (LSST DESC) et al., 2021).

O DP0.2 foi criado com o objetivo de gerar catálogos com um conjunto mais completo de produtos de dados, como esperado para os futuros releases, para teste e validação dos pipelines científicos. O DP0.2 carrega as características de profundidade 5 anos de observações e cobertura de área de 300 *graus*^{2}, centrada em RA, DEC = 61.863, -37.790 graus. O LSST Science Pipelines segue 4 etapas principais e vamos descrevê-las para o DP0.2.

1. Processamento de imagem única: são feitas as correções instrumentais relacionadas às detecções nos CCDs (remoção de assinatura do instrumento).

2. Calibração: calibração astrométrica e fotométrica das imagens, baseadas em um catálogo (neste caso, simulado) de referência. Aqui são geradas as imagens calibradas chamadas de *calexps*.

3. Coadição de imagens: empilhamento das calexps para gerar as imagens mais profundas, chamadas de *coadd*. A grade de coadds é dividida em *tracts* e *patches*, para otimizar o processamento. Um patch tem 4100 x 4100 pixels de 0.2 arcsec cada, totalizando 13,7 arcmin de lado, que configura o tamanho do CCD. Um tract é formado

por 7x7 pacthes. As bordas dos pacthes se sobrepõem em 100 pixels e as bordas dos tracts se sobrepõem em 1 arcmin. Na figura 4.2, há uma ilustração dessa divisão.

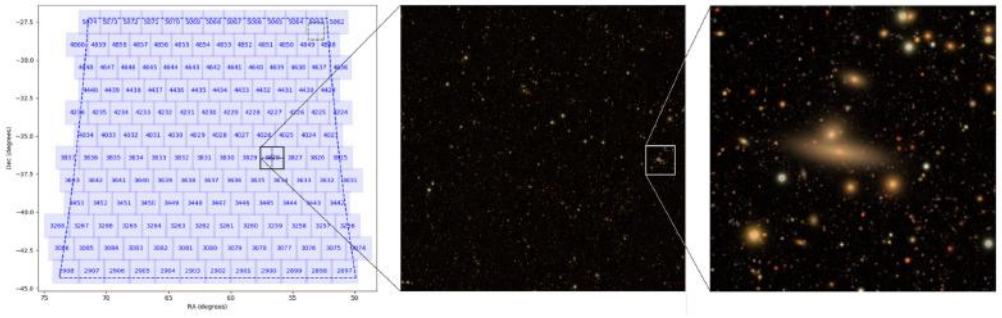


FIGURA 4.2. Ilustração da divisão dos dados em tracts e patches (LSST Dark Energy Science Collaboration (LSST DESC) et al., 2021)

4. Processamento de coadd: feitas as etapas para geração dos produtos de dados para os catálogos como as detecções em cada banda, agrupamento das detecções de cada banda correspondentes (diferentes detecções do mesmo objeto), aplicação do deblending - técnica avançada para separação dos fluxos de objetos sobrepostos, observados na mesma linha de visada (Melchior et al., 2018)-, escolha da banda de referência e aplicação da fotometria forçada.

Uma descrição completa de todas as tabelas de produtos de dados geradas para o DP0.2 podem ser encontradas em `https://sdm-schemas.lsst.io/dp02.html`.

4.2 Construção do *training set*

4.2.1 Amostra inicial

Inicialmente, temos um total de 278.318.455 de objetos na tabela Object, que contém as propriedades dos objetos detectados nas imagens coadd, entre estrelas, galáxias, supernovas e potenciais objetos espúrios. São fornecidos os fluxos e erros de fluxos dos objetos e não há correção de extinção. Esses dados podem ser acessados no Rubin Science Plataform, por meio de buscas com o ADQL, mas também se encontram alocados no cluster Apollo do LIneA, acessados pelo ambiente HPC. Para os objetivos deste trabalho, o acesso pelo ambiente HPC do LIneA foi mais viável pela demanda computacional do volume de dados.

O primeiro passo foi construir a *skinny table*, como descrito na seção 3.3. O script de geração da *skinny table* abrange além da seleção das colunas de interesse, a conversão de fluxo para magnitude, correção de extinção, tratamento de dados inválidos, e aplicação da flag detect_isPrimary = True, que seleciona objetos fora de áreas de sobreposição de coordenadas (RA, DEC) nos tracts e patches, fornecendo um catálogo de objetos únicos. As colunas de interesse foram as ra, dec, magnitudes em cada banda e seus respectivos erros, e as flags de qualidade de interesse, detalhadas na próxima seção. O número de objetos na skinny table com essa seleção foi de 153.934.600 objetos.

4.2.2 Cortes de qualidade

Definimos aqui como *training set* o conjunto que será selecionado para divisão posterior em conjunto de treino e conjunto de teste para gerar e validar o modelo. Para selecionar os objetos de interesse, fizemos uma série de cortes e aplicação de *flags* de qualidade na *skinny table*, dentre eles seleção de galáxias, limites de detecção, cortes em S/N, flags de fotometria. Todas as aplicações estão listadas, com suas respectivas descrições, na Tabela 4.1. Após os cortes, temos 26.366.000 objetos na *skinny table* final.

Nas Figuras 4.3 e 4.4, temos as distribuições de magnitudes e a distribuição espacial de todos os objetos da *skinny table* gerada após os cortes.

Tabela 4.1. Cortes de qualidade aplicados no training set.

Corte/Flag	**Valor**	**Descrição**
refExtendedness	1	Seleção de galáxias
deblend_skipped	False	Objetos que não foram ignorados pelo deblend
i_cModel_flag	$< 10^{(-0.375)}$	Retirar objetos muito contaminados por vizinhos
mag_i	< 24.5	Corte na magnitude limite na banda i
mag_i	> 17	Limite de detecção na banda i
S/N	> 10 na banda i e > 5 em pelo menos duas bandas entre g/r/z/y	Cortes de S/N para cada galáxia
i_pixelFlags_clippedCenter	False	Centro da fonte próximo a um pixel ruim
i_pixelFlags_crCenter	False	Raio cósmico no centro da fonte
i_pixelFlags_edge	False	Fonte fora da região de exposição utilizável
i_pixelFlags_interpolatedCenter	False	Pixel interpolado no centro da fonte
i_pixelFlags_saturatedCenter	False	Pixel saturado no centro da fonte
i_pixelFlags_suspectCenter	False	Centro da fonte próximo a um pixel suspeito
i_pixelFlags_offimage	False	Centro da fonte fora do campo da imagem
i_pixelFlags_bad	False	Pixel ruim no footprint
i_pixelFlags_clipped	False	Pixel descatado no footprint
i_pixelFlags_cr	False	Raios cósmicos no footprint
i_pixelFlags_interpolated	False	Pixel interpolado no footprint
i_pixelFlags_saturated	False	Pixel saturado no footprint
i_pixelFlags_suspect	False	Pixel suspeito no footprint

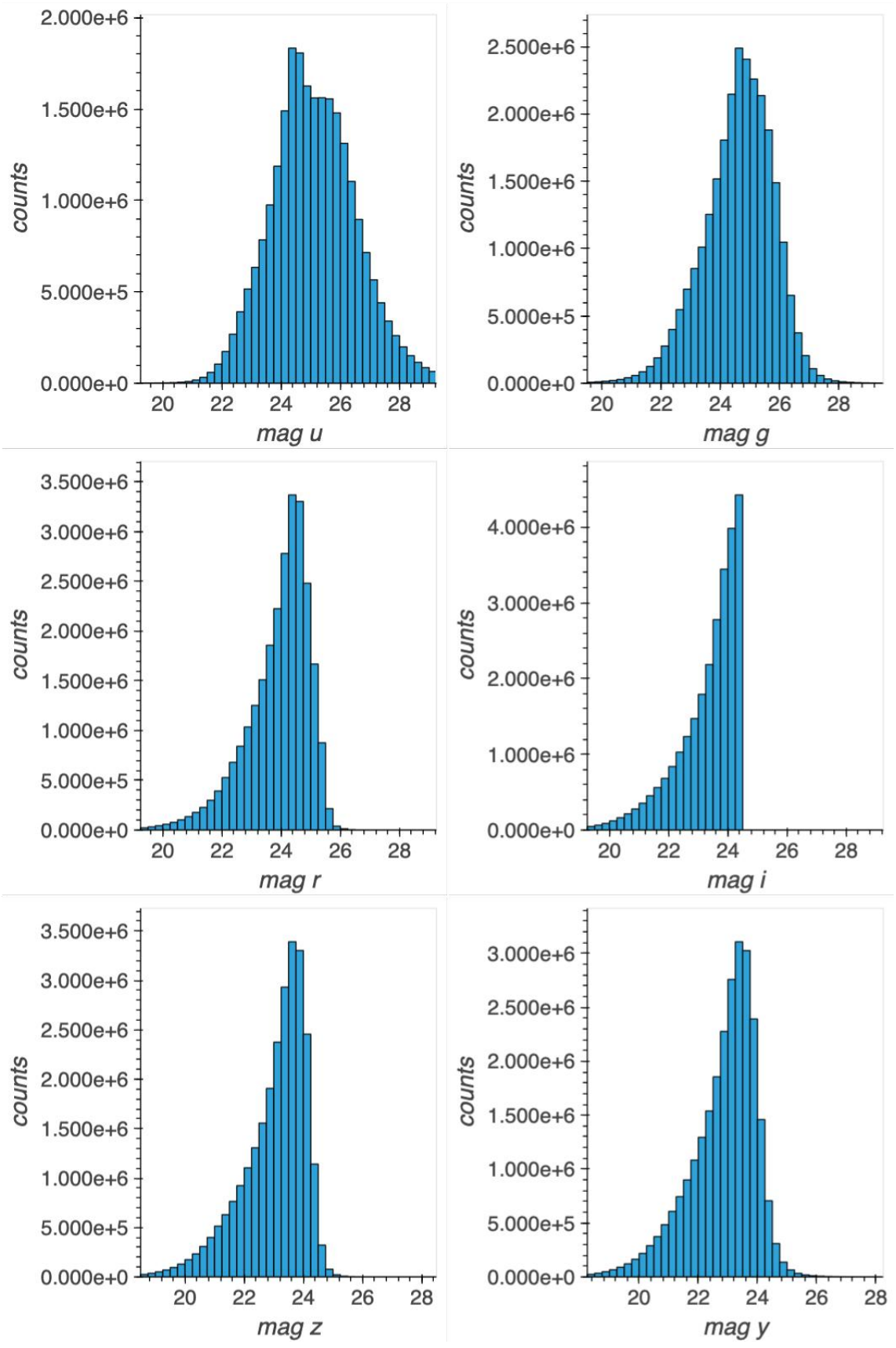


FIGURA 4.3. Distribuição de magnitudes em cada banda para todos os objetos da skinny table

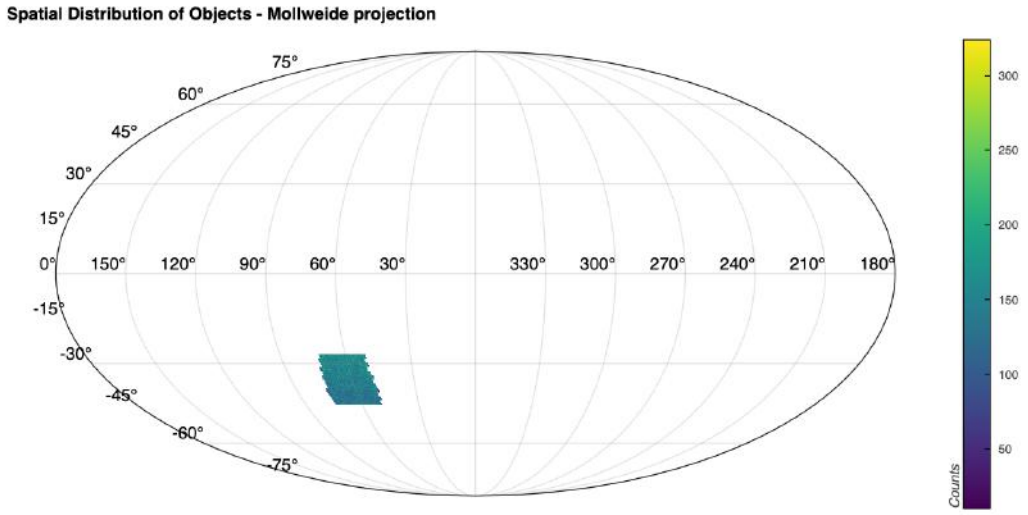
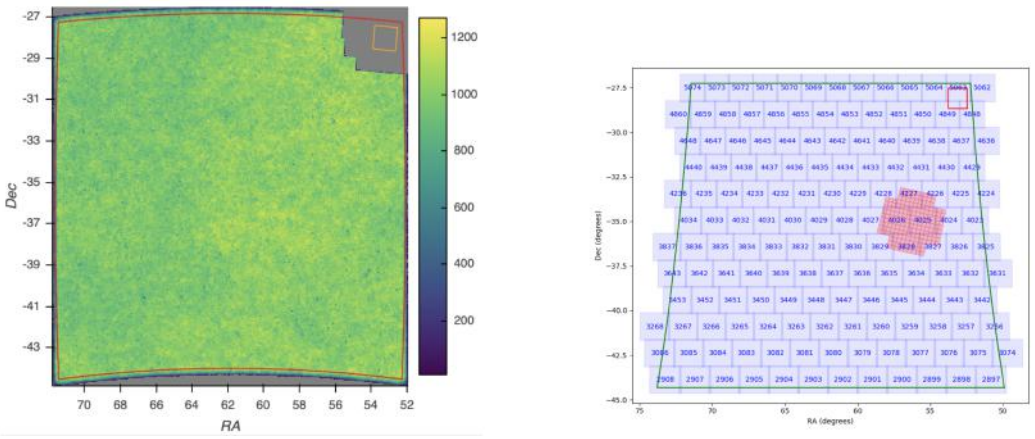


FIGURA 4.4. Distribuição espacial para todos os objetos da skinny table

Após aplicar os cortes à *skinny table*, selecionamos uma fração aleatória dos objetos filtrados pelos cortes de qualidade. Escolhemos arbitrariamente criar um *training set* com aproximadamente 100000 objetos, de forma que tivéssemos um conjunto de teste com tamanho razoável para o cálculo das estatísticas de qualidade. correspondente a aproximadamente 0.038% dos objetos iniciais da skinny table, sendo um total de 100.038 objetos. Após selecionar os objetos do *training set*, seguimos para a obtenção dos redshifts espectroscópicos de cada galáxia. A coluna de redshift está na tabela TruthSummary do DP0.2, que contém um resumo das propriedades da tabela-verdade da simulação DESC DC2. Fizemos um cross-matching espacial, utilizando a biblioteca LSDB, através de Jupyter notebooks desenvolvidos pelo LIneA, referenciados no Apêndice B, aplicados no ambiente HPC.

Na simulação DC2, algumas regiões foram simuladas com diferentes estratégias de observação. Na Figura 4.5, temos a região que foi simulada com a estratégia WFD do LSST, onde temos o mapeamento extensivo do céu com cadência padrão, deixando de fora as bordas da região coberta pela simulação. Escolhemos seguir apenas com essa região no *training set* para garantir o cross-matching entre objetos com mesma estratégia de observação. Há uma pequena região também sinalizada na Figura 4.5 que foi utilizada para simular a estratégia de DDP, onde uma pequena região é usada para fazer observações mais profundas. Os objetos desses tracts não foram utilizados na produção das imagens *coadds* para manter a uniformidade dos dados, por isso não há

objetos nessa região para o crossmatching, e esses tracts não aparecem na distribuição espacial final do *training set*.



(A) (B)

FIGURA 4.5. Região WFD para a simulação DC2. Na figura A, temos a projeção Mollweide para a região WFD; o quadrado laranja no canto superior direito indica a região DDP, onde observamos a ausência dos tracts. Na figura B, temos a região WFD destacada pelo trapézio verde na projeção equatorial, com a região DDP novamente destacada pelo quadrado vermelho; a região listrada representa o plano focal da LSSTCam (LSST Dark Energy Science Collaboration (LSST DESC) et al., 2021).

Tratamos as não-detecções como *NaNs*, uma vez que o algoritmo consegue lidar com elas de forma interna. Por fim, fizemos a divisão do *training set* conforme descrito na seção 3.4, sendo 70.216 objetos para o conjunto de treino e 29.912 para o conjunto de teste.

A figuras a seguir correspondem à área do céu (Fig. 4.6), distribuição de magnitudes (Fig. 4.7), erros das magnitudes (Fig. 4.8), diagrama cor-cor (Fig. 4.9), magnitude x cor (Fig. 4.10), distribuição de redshift (Fig. 4.11) e magnitude x redshift (Fig. 4.12) para os objetos do *training set*.

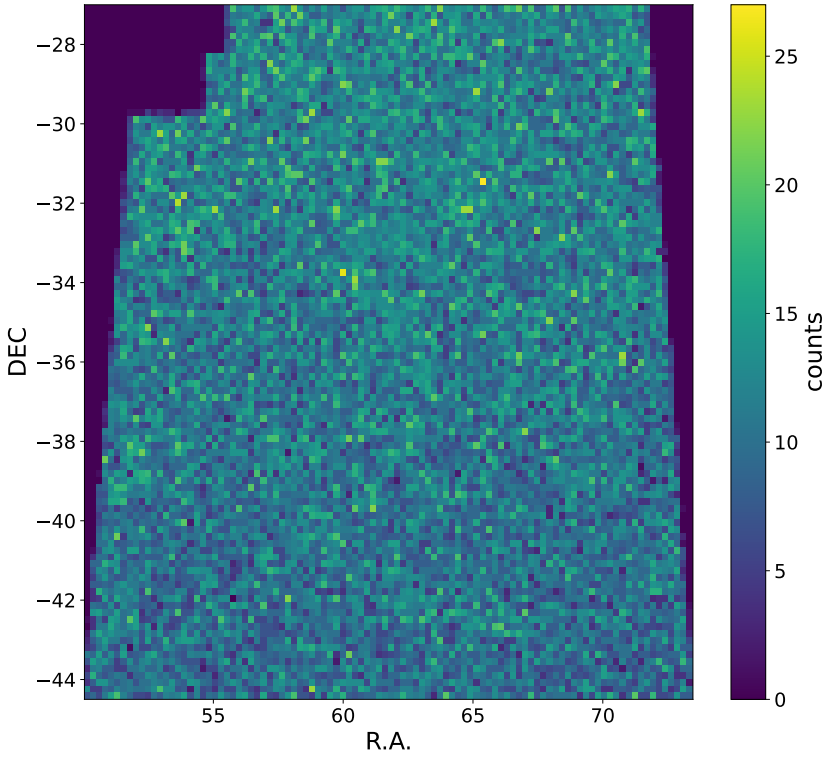


FIGURA 4.6. Área do céu para os objetos do *training set*.

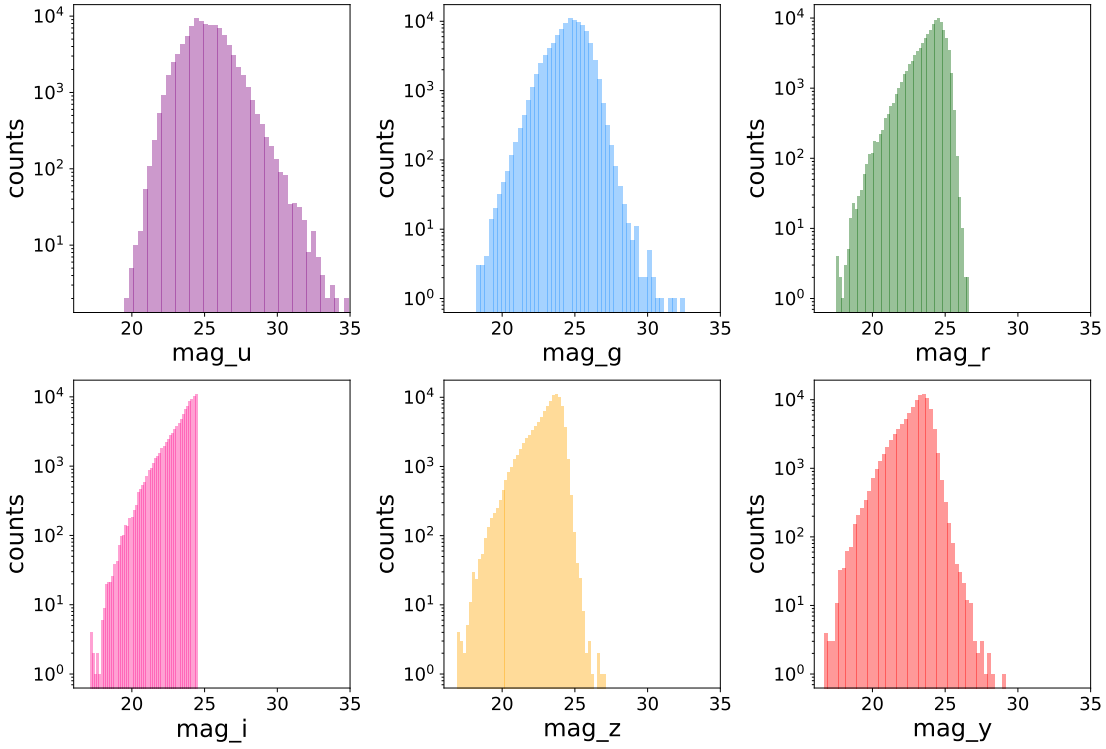


FIGURA 4.7. Distribuição de magnitudes para os objetos do *training set*.

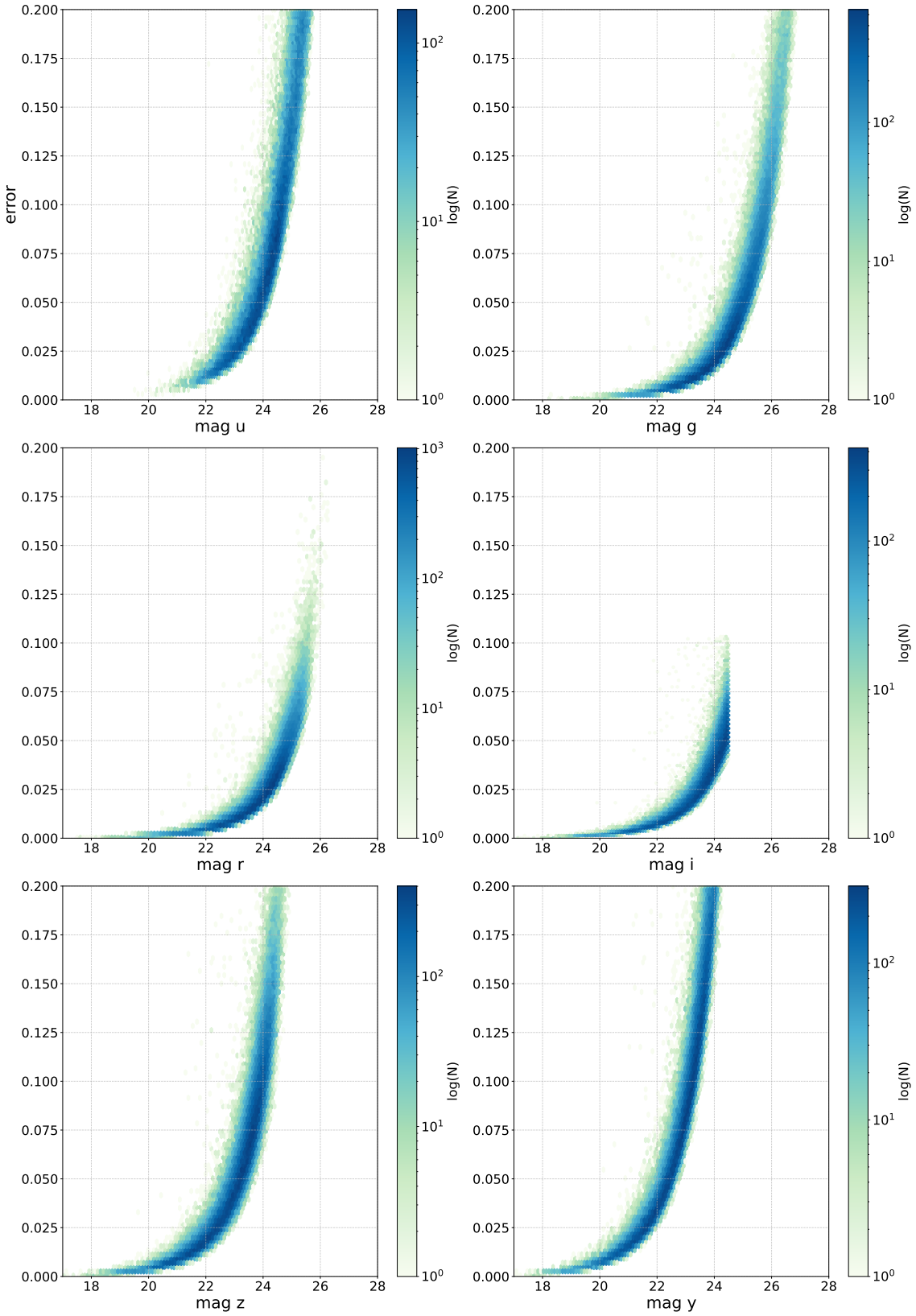


FIGURA 4.8. Erros das magnitudes dos objetos do *training set*.

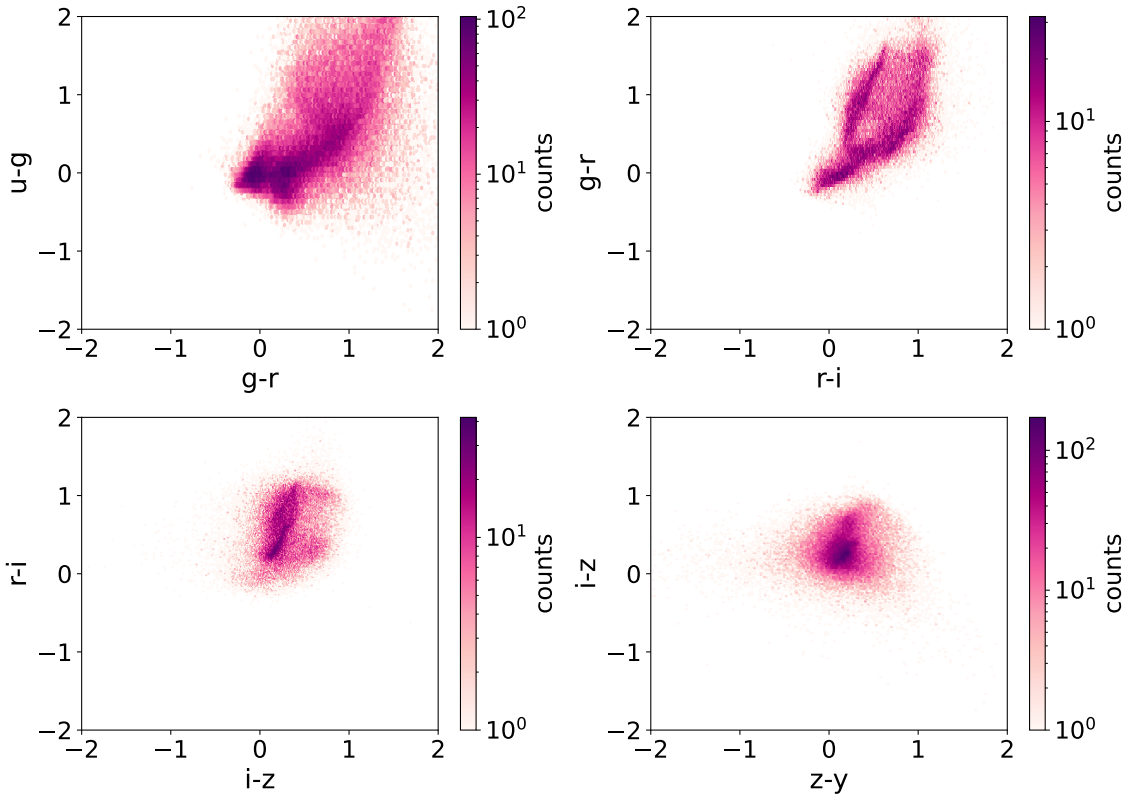


FIGURA 4.9. Diagramas cor-cor.

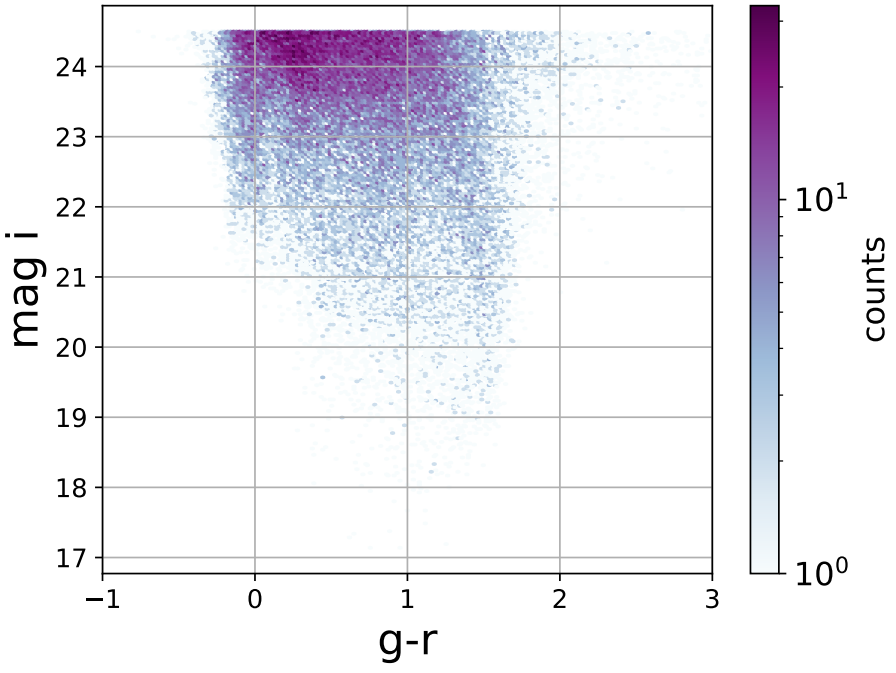


FIGURA 4.10. Magnitudes na banda i por cor g-r.

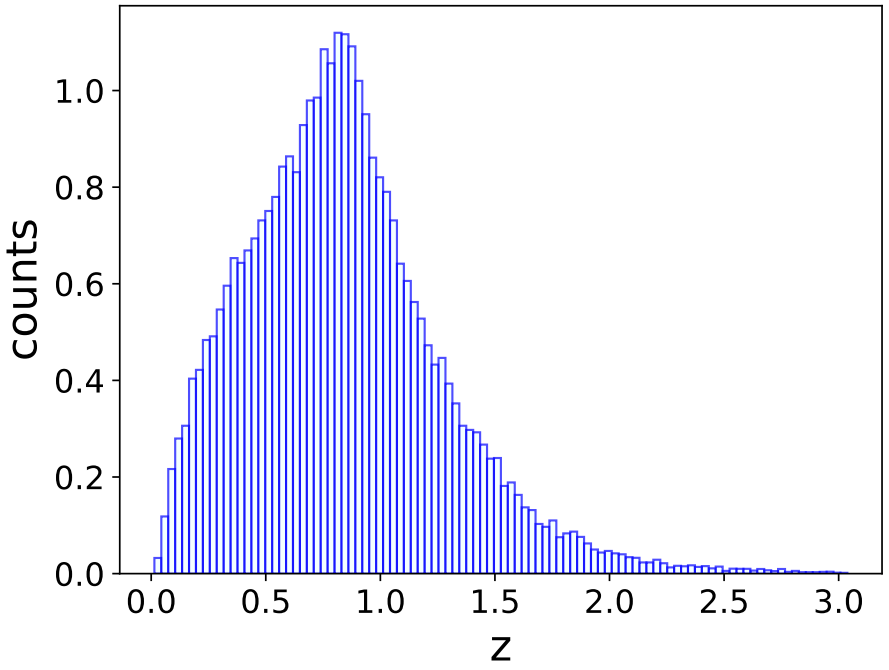


FIGURA 4.11. Distribuição de redshifts para os objetos do *training set*.

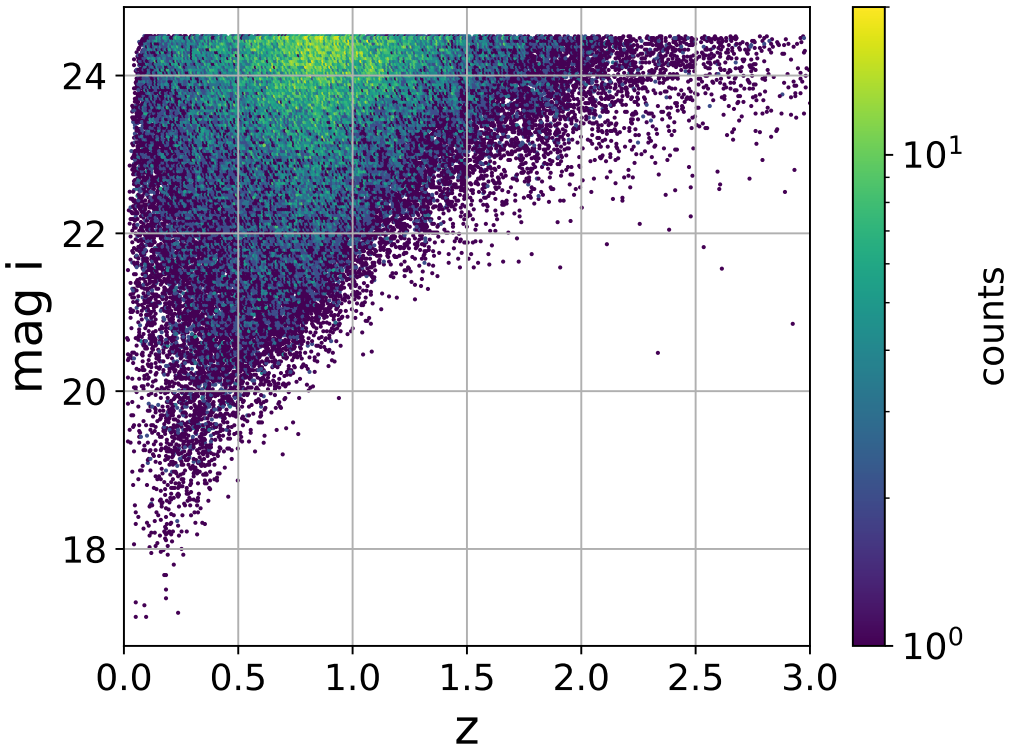


FIGURA 4.12. Magnitudes por redshift na banda i.

Capítulo 5

Resultados e discussões

Neste capítulo, realizamos a avaliação do desempenho do modelo no conjunto de teste. Inicialmente, analisamos o impacto dos principais hiperparâmetros e da seleção de atributos nas métricas de desempenho, com o objetivo de definir a configuração final do modelo. Em seguida, apresentamos as métricas completas obtidas para o modelo final, a distribuição das estimativas de photo-zs na tabela completa, bem como a análise do desempenho técnico do algoritmo.

5.1 Hiperparâmetros e atributos

Um dos hiperparâmetros mais importantes a ser definido na construção da floresta aleatória é a quantidade de árvores da mesma. De acordo com Carrasco Kind & Brunner (2013), as métricas não sofrem alterações significativas para mais de 100 árvores, sendo esse um número razoável para a construção da floresta. A Figura 5.1 mostra a variação das métricas para o nosso conjunto de teste conforme variamos o número de árvores. A queda brusca observada no início da curva evidencia a diferença entre usar apenas uma árvore e aplicar uma floresta aleatória, confirmando a eficácia deste método. A partir de 20 árvores, as métricas diminuem gradualmente, apresentando um valor um pouco mais baixo a partir de 100 árvores, em comparação aos outros valores.

Em conjunto com esses dois resultados, seguimos com o número de árvores igual a 100.

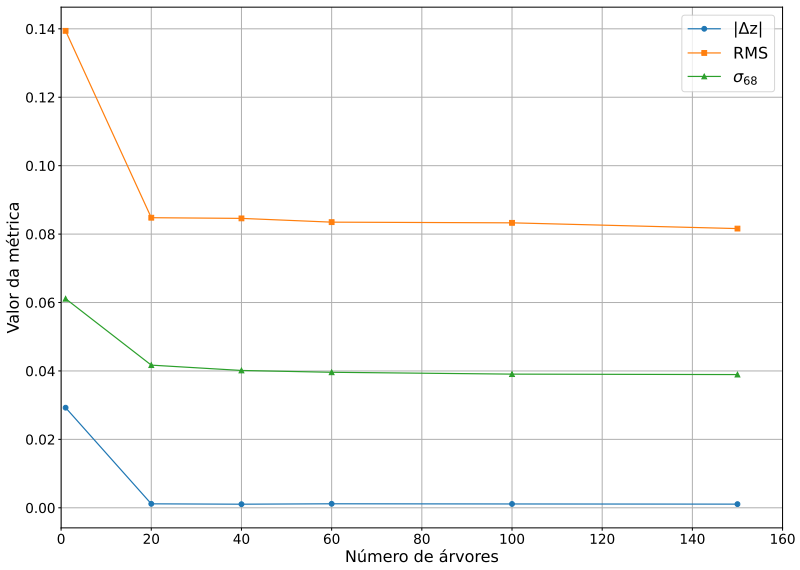


FIGURA 5.1. Variação das métricas em função do número de árvores.

Outro hiperparâmetro de grande importância é o $minleaf$, que determina a profundidade de cada árvore da floresta. Observamos que, ao aumentar seu valor, algumas regiões específicas do gráfico de distribuição dos photo-zs apresentavam picos maiores, indicando regiões fortes de outliers. Esses picos aparecem devido à degenerescência nas cores das galáxias, causada quando a quebra de 4000 Å se posiciona entre duas bandas, o que gera outliers na estimativa nessas regiões. A Figura 5.2 ilustra, como exemplo, esse comportamento nos filtros do DES. Conforme aumentamos o valor do $minleaf$, teremos menos galáxias na folha terminal da árvore e isso significa que o modelo capturou menos informação sobre o conjunto, se tornando mais suscetível a errar as predições. O hiperparâmetro por si só não tem ligação com a física do problema, porém é interessante notar como a forma com que o algoritmo é estruturado pode influenciar nos resultados científicos.

Pensando, então, na informação que o algoritmo recebe, testamos também como alterar os atributos impactaria na quantidade de outliers nesta região ao variar o $minleaf$. Testamos utilizar como atributos as magnitudes $ugrizy$, as cores $u-g$, $g-r$, $r-i$, $i-z$ e $z-y$, e a junção das magnitudes e cores. Este último pode parecer redundante, porém a árvore de decisão é menos sensível a essas redundâncias entre os atributos, uma vez que ela escolhe qual atributo separa melhor os dados naquele momento, o que facilita a captura de informações relevantes durante as divisões.

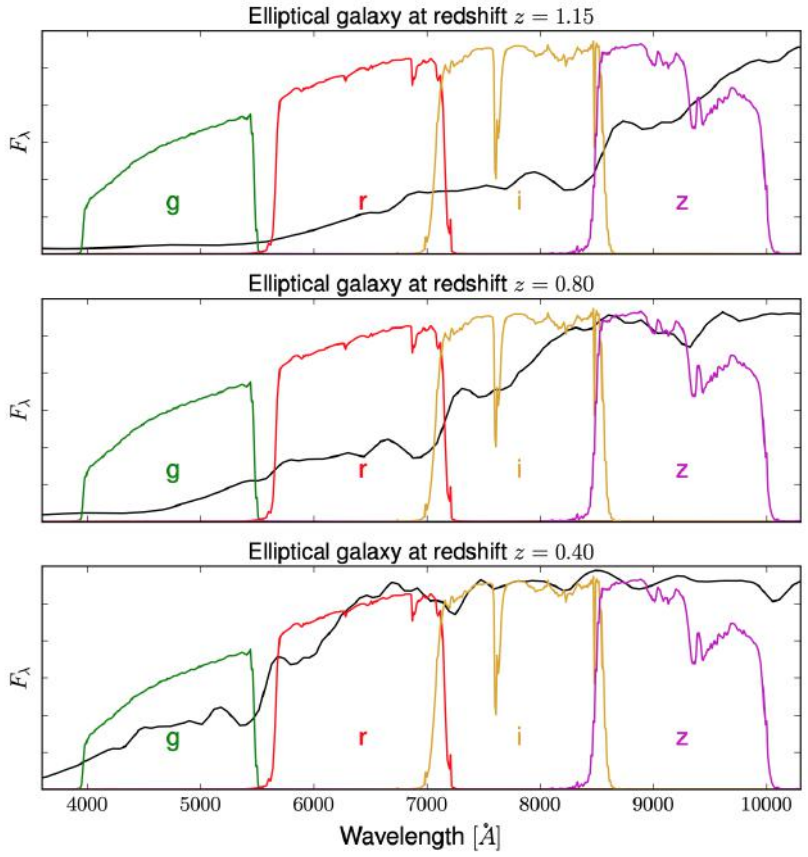
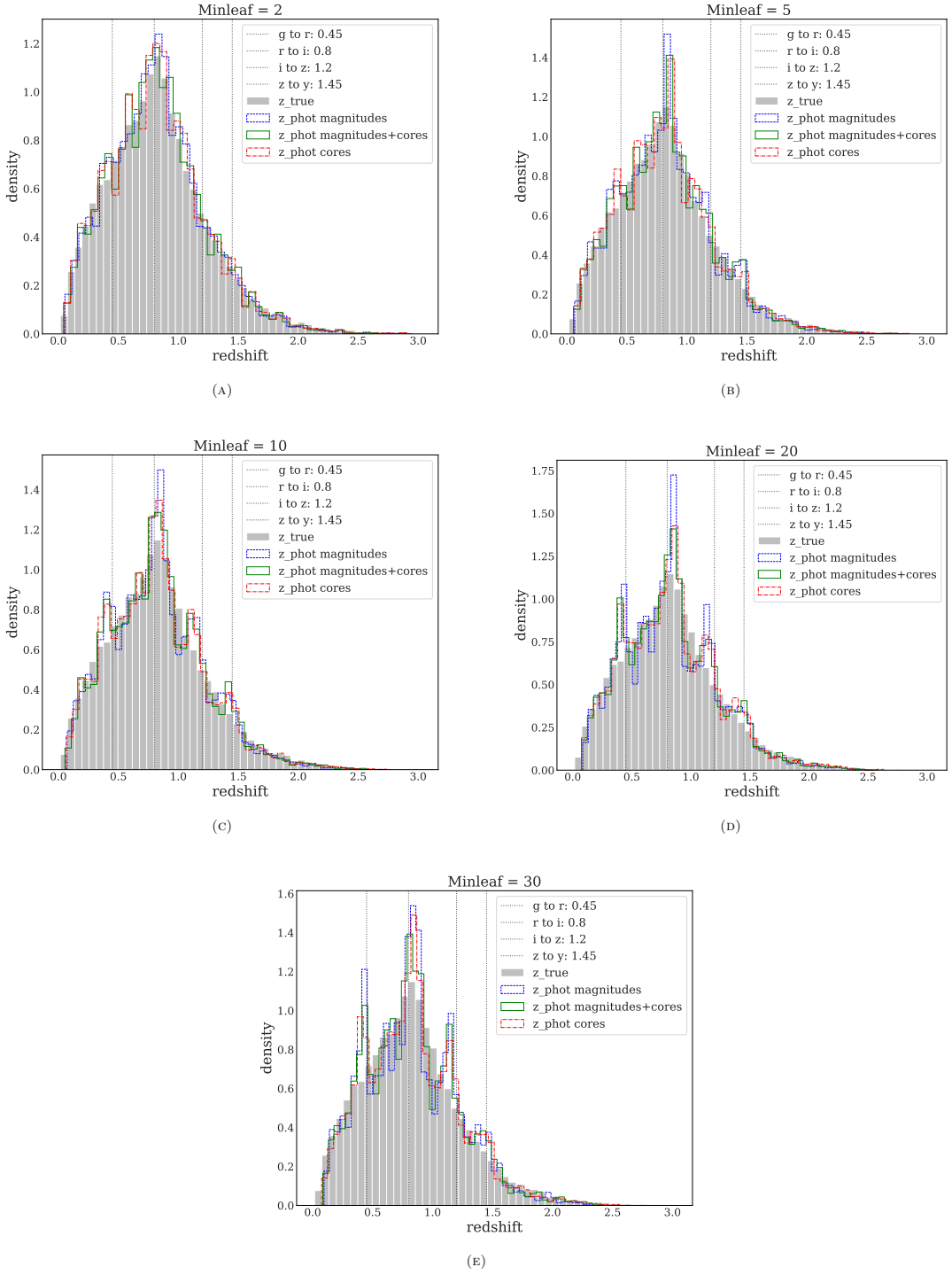


FIGURA 5.2. Posicionamento da quebra de 4000 Å entre bandas passantes (Crocce et al., 2019).

À medida que aumentamos o *minleaf*, os picos citados anteriormente se intensificam e a diferença entre os picos para cada escolha de atributos fica mais evidente. Utilizar as cores nos atributos mostra uma melhora na intensidade dos picos em algumas regiões, porém não é suficiente para que eles diminuam ao ponto que justifique trocar as magnitudes pelas cores ou adicioná-las para treinar o algoritmo, de forma que as métricas gerais não sofrem alterações significativas. Observamos que um valor de *minleaf* menor traz uma distribuição de photo-zs mais semelhante à distribuição real e, nesse regime, não há diferença em escolher magnitudes ou cores como atributos, a distribuição de galáxias por redshift é semelhante. Decidimos seguir, então, com as magnitudes. A Figura 5.3 ilustra os resultados dessa discussão.

Diante disso, a questão é que uma escolha pequena de galáxias na folha terminal pode indicar um *overfitting*, pois o algoritmo tende a se ajustar demais aos dados do treinamento. Para investigar este ponto, fizemos a avaliação do desempenho do modelo nos conjuntos de treino e validação, através da métrica RMSE, que avalia o erro médio das predições. Notamos que, de fato, para $minleaf = 2$ o modelo apresentou um forte



(A) (B) (C) (D) (E)

FIGURA 5.3. Distribuição dos redshifts fotométricos em função do $minleaf$, de acordo com cada escolha de atributos. A escolha do $minleaf$ em cada rodada está indicada no título de cada plot. As linhas verticais indicam as faixas de passagem de filtro para os filtros do LSST.

overfitting inicialmente, mas esse efeito foi reduzido ao variarmos o parâmetro *nrandom*. Ao criar novos catálogos a partir do catálogo original, com dispersão gaussiana, o modelo é exposto a diferentes cenários para os mesmos objetos, o que aumenta seu poder de generalização, permitindo usar um *minleaf* baixo sem causar overfitting, o que está de acordo com o que Carrasco Kind & Brunner (2013) descreve sobre os catálogos introduzirem aleatoriedade nas árvores. A Figura 5.4 mostra a variação do RMSE de treino e teste em função do aumento do *nrandom*. Com base nesse resultado, escolhemos $nrnadom = 10$ e $ntree = 10$.

Os demais hiperparâmetros foram definidos com base nos resultados de Carrasco Kind & Brunner (2013) e do notebook de exemplo Schmidt (2025). A configuração completa do algoritmo está descrita no Apêndice B.

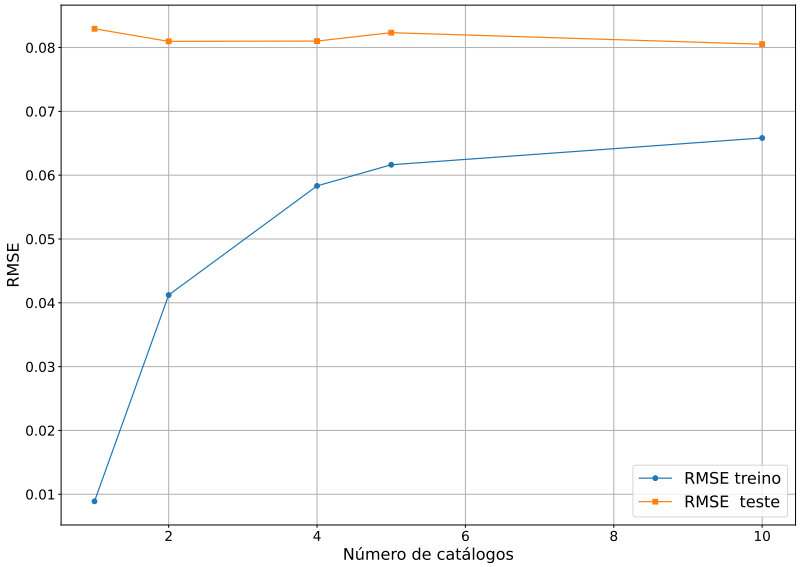


FIGURA 5.4. Validação cruzada, mostrando a variação do RMSE de treino e validação em função do hiperparâmetro *nrandom*.

5.2 Métricas do conjunto de teste

5.2.1 Descrição das métricas

As métricas utilizadas para avaliação do modelo foram:

1. Δz: quantifica o erro em uma estimativa de redshift fotométrico em relação ao redshift espectroscópico

$$\Delta z = \frac{z_{phot} - z_{spec}}{1 + z_{spec}}; \tag{5.1}$$

2. RMSE: Mede a dispersão global dos erros nas estimativas de redshift fotométrico, sendo calculado como

$$RMSE = \sqrt{\langle (\Delta z)^2 \rangle}; \tag{5.2}$$

3. σ_{68}: dispersão das estimativas de redshift fotométrico em relação aos valores reais espectroscópicos, dentro de uma faixa que contém 68% dos objetos da amostra;

4. $out_{2\sigma}$: fração de objetos em que o erro de redshift fotométrico (Δz) é maior que 2 vezes o desvio padrão (2σ) em relação ao valor real do redshift espectroscópico;

5. $out_{3\sigma}$: fração de objetos em que o erro de redshift fotométrico (Δz) é maior que 3 vezes o desvio padrão (3σ) em relação ao valor real do redshift espectroscópico.

6. PIT-QQ: A transformação integral de probabilidade (PIT, do inglês *Probability Integral Transform*) leva uma variável aleatória contínua (estimativa de Z_{phot}) a uma nova variável aleatória com distribuição uniforme no intervalo [0,1], a partir de sua função de distribuição cumulativa (obtida a partir da PDF gerada). Neste plot, são apresentados o histograma com os valores PIT e o Quantil-Quantil(QQ), que compara os quantis das distribuições de z_{spec} e z_{phot}. Quanto mais uniforme for o histograma e quanto mais próximo da linha diagonal for o QQ plot, melhor será a qualidade das PDFs e, consequentemente, mais robusto o modelo gerado (Schmidt et al., 2020). A Figura 5.5 mostra exemplos do PIT-QQ plot para vários algoritmos, incluindo o TPZ.

5.2.2 Avaliação das métricas no conjunto de teste

Como avaliação dos resultados da aplicação do modelo gerado no conjunto de teste, vamos visualizar e analisar as métricas de desempenho. A Figura 5.6 apresenta uma análise visual de quanto os valores de redshift estimados pelo modelo se dispersam, sendo o desejado o mais próximo possível da reta $y = x$. Acompanhando a barra lateral do gráfico, observamos que há uma grande densidade de objetos próximos à reta, ou

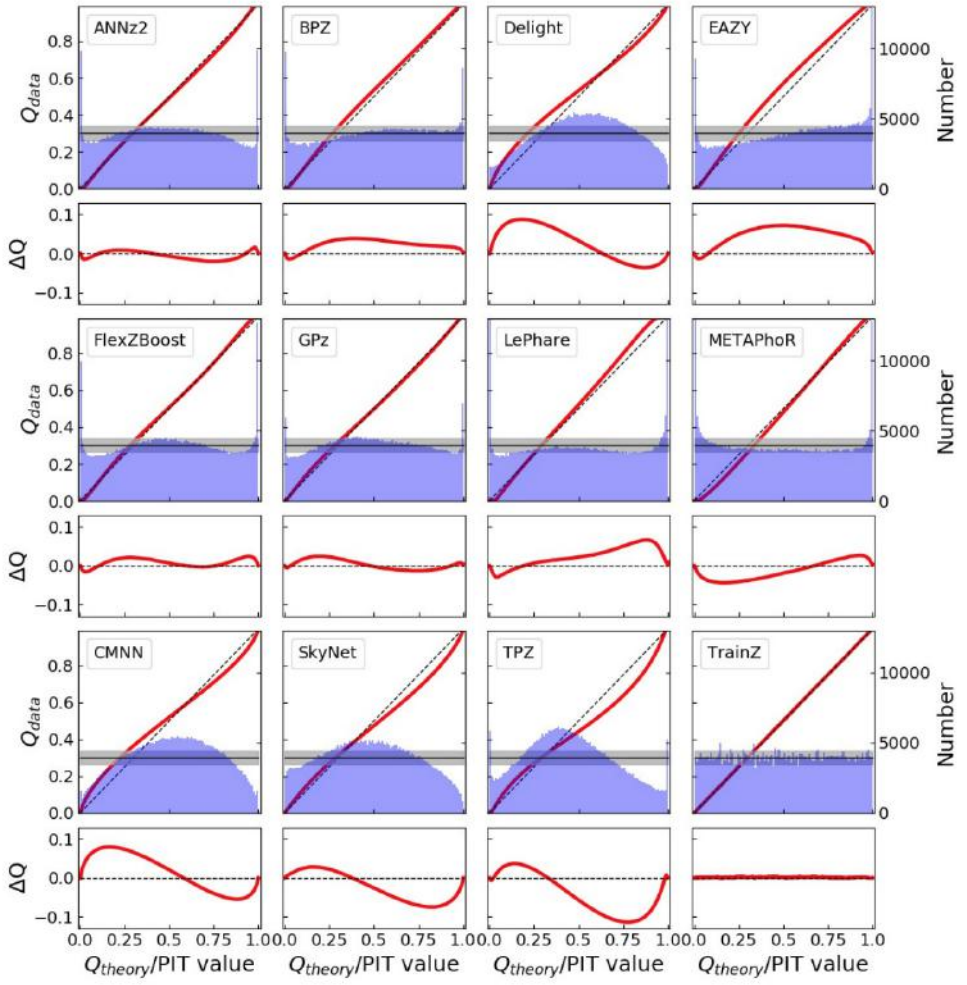


FIGURA 5.5. Exemplos de PIT-QQ plot para diferentes algoritmos de photo-z (Schmidt et al., 2020).

seja, a estimativa do photo-z se aproxima do valor real. Temos uma esperada fração de outliers, que será melhor discutida com a análise das métricas subsequentes.

Na Tabela 5.1, temos os resultados das métricas para nosso conjunto de teste. Vemos que o modelo apresenta boa predição na média, com viés bem próximo de zero. Temos ainda uma baixa fração de outliers e boa dispersão central. O RMSE um pouco mais alto indica que há erros maiores quando olhamos para as previsões de forma individual, mas ainda apresenta um valor aceitável, considerando que temos um erro sistemático baixo.

Na Figura 5.7, temos os valores de bias, σ_{68} e frações de outliers por faixa de redshifts. Vemos que os valores se tornam maiores conforme se aproximam de redshifts mais altos, o que é esperado. Para esses valores de redshift, temos mais incertezas

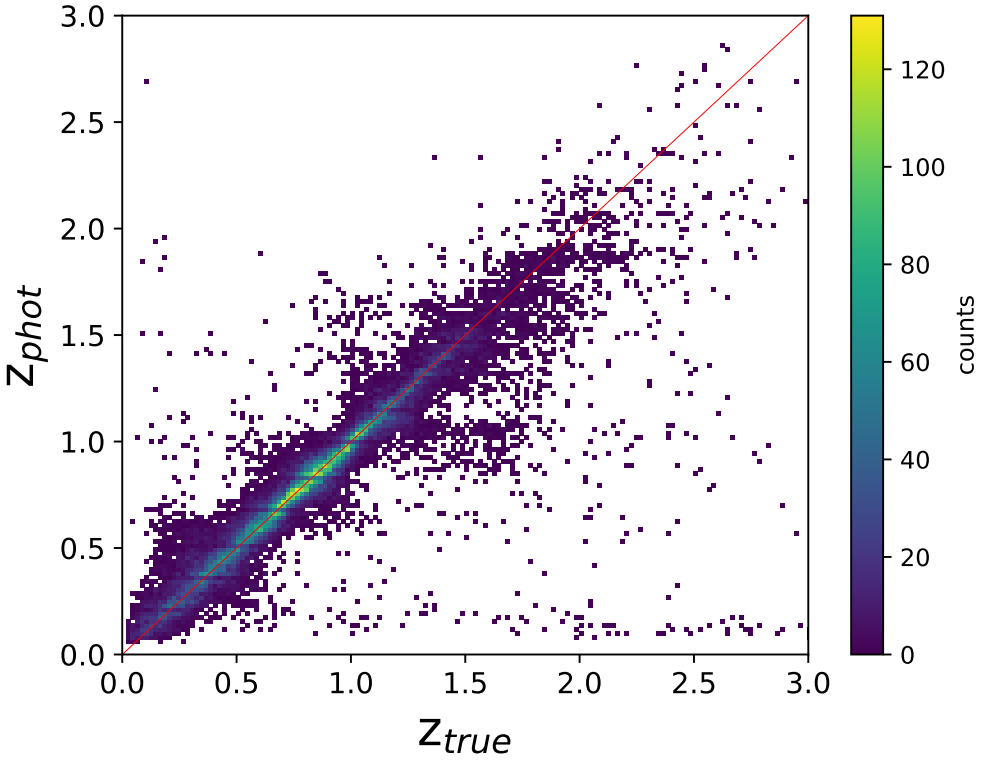


FIGURA 5.6. Gráfico de dispersão de z_{phot} x z_{true}

fotométricas, além de um desbalanceamento na quantidade de galáxias nessa faixa, o que dá ao algoritmo menos informações para fazer predições corretas.

TABELA 5.1. Métricas de desempenho do redshift fotométrico.

Métrica	**Desempenho do modelo**
Bias	-0.0002
RMS	0.079
σ_{68}	0.038
Fração de outliers $> 2\sigma$	2%
Fração de outliers $> 3\sigma$	4%

Como descrito na seção 3.1, as saídas da estimativa de photo-z são as PDFs de redshift para cada galáxia. Trouxemos aqui alguns exemplos de PDFs individuais, exibidas na Figura 5.8. Para estimativas acuradas do photo-z, esperamos PDFs de largura coerente com as incertezas e pico próximo do valor verdadeiro de redshift. PDFs muito largas ou ruidosas mostram grande incerteza na medição do redshift. As figuras exibidas na primeira linha mostram PDFs boas, com formato coerente e com valor mais provável para o photo-z, que é o valor do pico da PDF, aproximadamente igual ao valor real. Na segunda linha, vemos PDFs com boas predições, porém com formatos mais ruidosos, o que aumenta as incertezas da medição. Por fim, na última linha, vemos PDFs muito

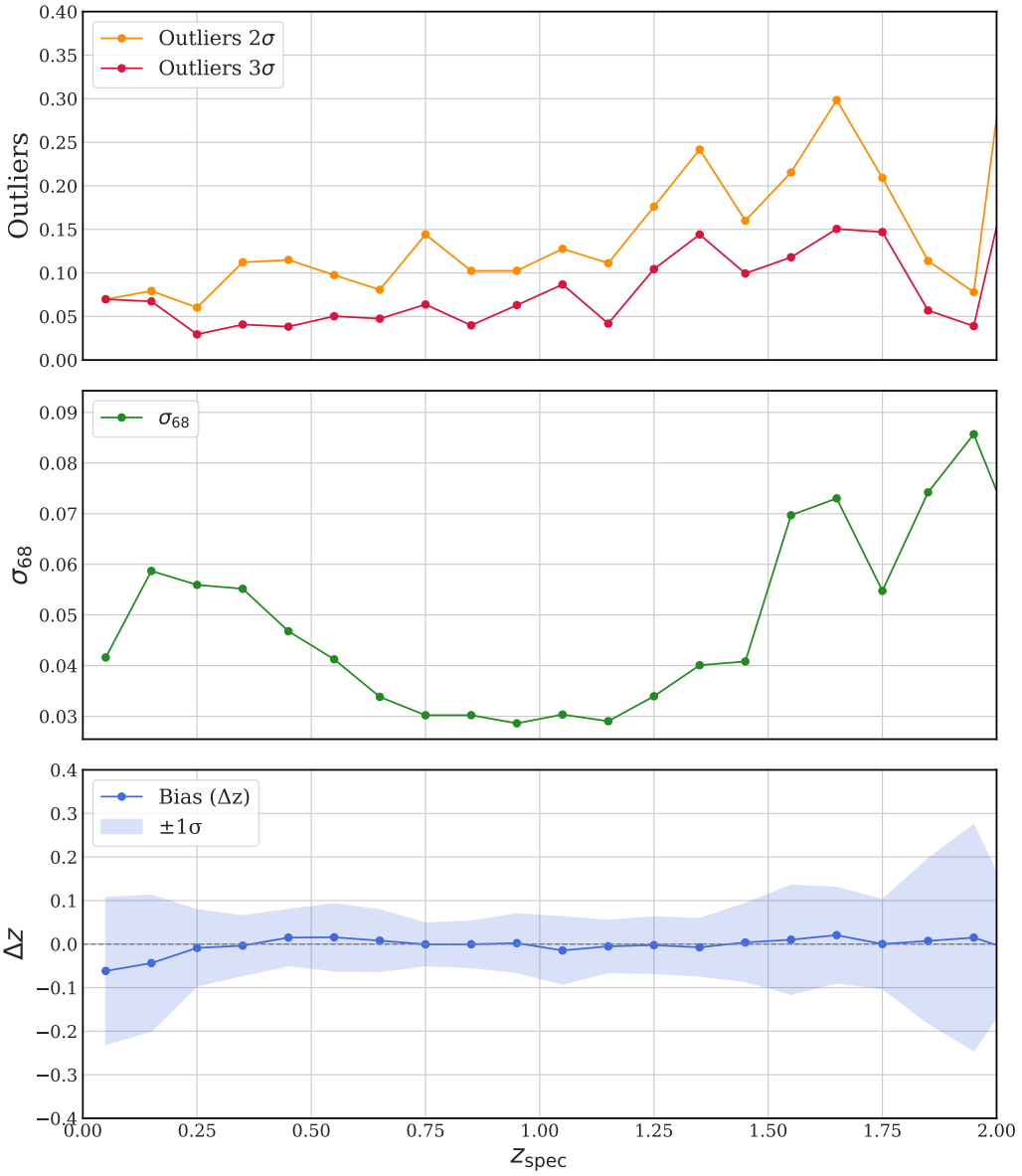


FIGURA 5.7. Métricas de dispersão e viés. Acima σ_{68},$out_{2\sigma}$ e $out_{3\sigma}$, e abaixo Δz.

ruidosas e com predições ruins, onde o valor mais provável do photo-z está distante do valor real. Essas PDFs são exemplos de photo-zs outliers das regiões de passagem de filtro, conforme discutido anteriormente.

Como análise geral da calibração das PDFs geradas pelo nosso modelo, apresentamos na Figura 5.9 o gráfico PIT-QQ. Observa-se um desvio da linha vermelha em relação à diagonal, indicando viés na distribuição prevista. O histograma do PIT também apresenta comportamento não uniforme, reforçando essa evidência. O valor de PIT_{out} quantifica esse desvio, medindo a proporção de objetos com valores extremos de PIT; para uma distribuição uniforme ideal, o valor de PIT seria de 0.0002. No nosso

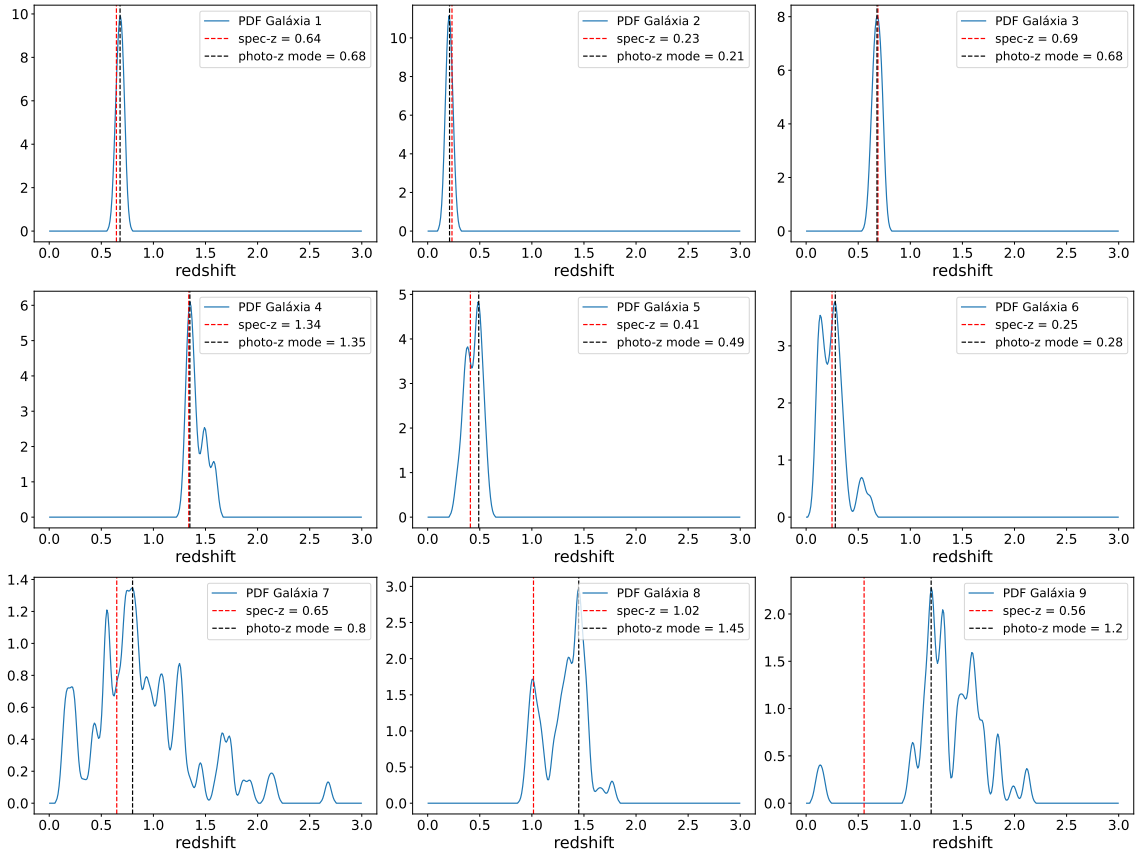


FIGURA 5.8. Exemplos de PDFs individuais para o conjunto de teste.

caso, o valor obtido foi $PIT_{out} = 0.0252$, que indica uma fração considerável de outliers catastróficos (Schmidt et al., 2020))=. Apesar disso, o modelo apresenta desempenho satisfatório na predição da maioria dos objetos, sugerindo que, embora melhorias na calibração das PDFs sejam desejáveis, os resultados gerais permanecem razoáveis.

Por fim, comparamos a distribuição de galáxias por redshift prevista com a real, chamada $N(z)$. O $N(z)$ foi construído de duas maneiras: a primeira utiliza um histograma dos valores z_{mode}, que correspondem aos valores mais prováveis estimados para cada redshift fotométrico, ilustrado na Figura 5.10; a segunda forma considera a soma das PDFs individuais, resultando em uma PDF global que representa a distribuição de probabilidade $p(z)$ contínua de redshifts da amostra, ilustrada na Figura 5.11.

Conseguimos observar que a predição feita no conjunto de teste se aproxima da distribuição real, de acordo com as métricas vistas anteriormente. No histograma dos valores de redshift na Figura 5.11, vemos que os picos mais distantes do original se mostram de fato nas regiões de passagem de filtro, como já discutido.

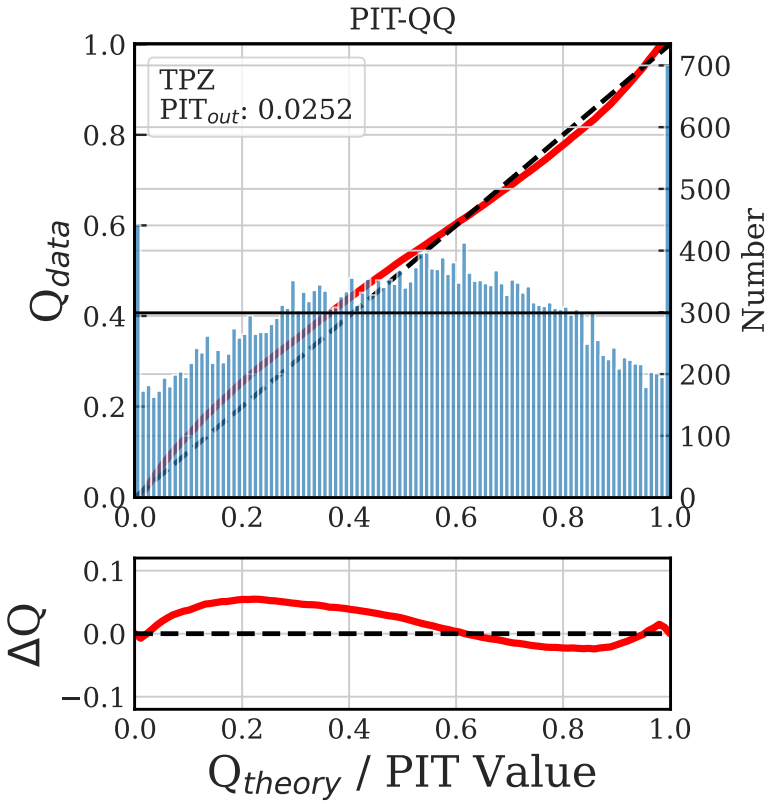


FIGURA 5.9. PITQQ plot

Comparando a Figura 5.10 com a soma das PDFs na Figura 5.11, há uma nítida suavização na curva. Isso se dá porque os picos das PDFs individuais, espalhados ao longo do eixo do redshift, se cancelam parcialmente ao serem empilhados.

Podemos comparar os resultados obtidos neste trabalho com os apresentados em Schmidt et al. (2020), no qual a performance de diversos algoritmos de estimativa de photo-zs foi avaliada utilizando um catálogo simulado representativo das condições observacionais do LSST após 10 anos de operação. Considerando as métricas tradicionais, observamos que a fração de outliers em nosso conjunto é ligeiramente superior à reportada por Schmidt et al. (2020) para o algoritmo TPZ, sendo de aproximadamente 4% no presente trabalho, contra 3% no estudo comparativo. O gráfico de dispersão apresenta um comportamento qualitativamente semelhante, indicando compatibilidade no desempenho geral. Ao analisarmos a distribuição dos photo-zs por meio do gráfico PIT-QQ, verificamos que nossa distribuição é mais uniforme ao longo da diagonal ideal, sugerindo uma melhor calibração global das distribuições preditivas. No entanto, o valor de PIT_{out} em nosso caso é superior à obtida por Schmidt et al. (2020), cuja taxa foi de

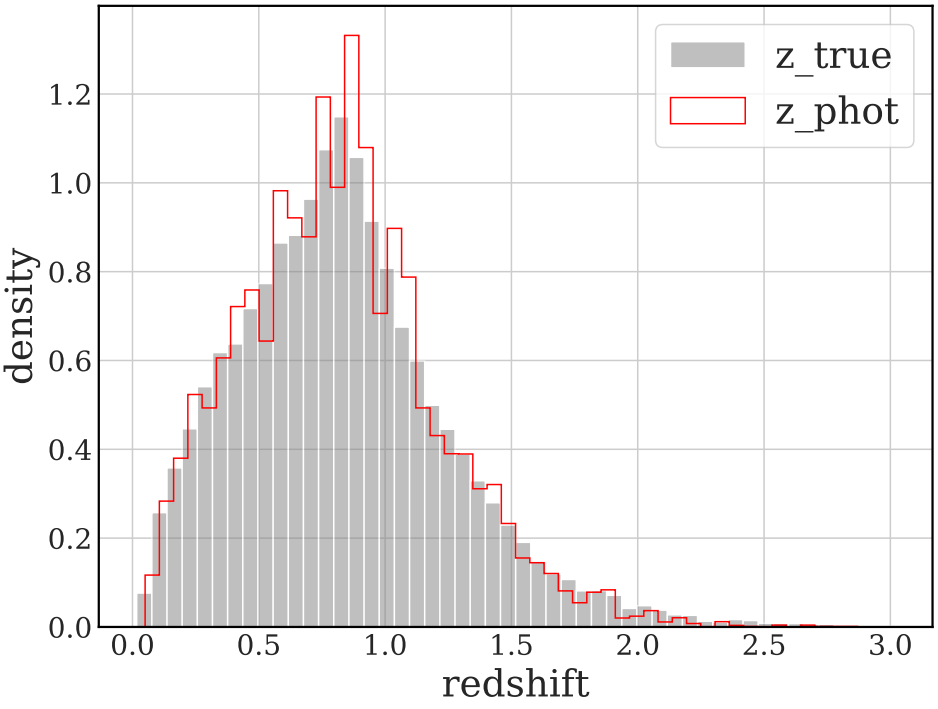


FIGURA 5.10. Histograma z true e zmode.

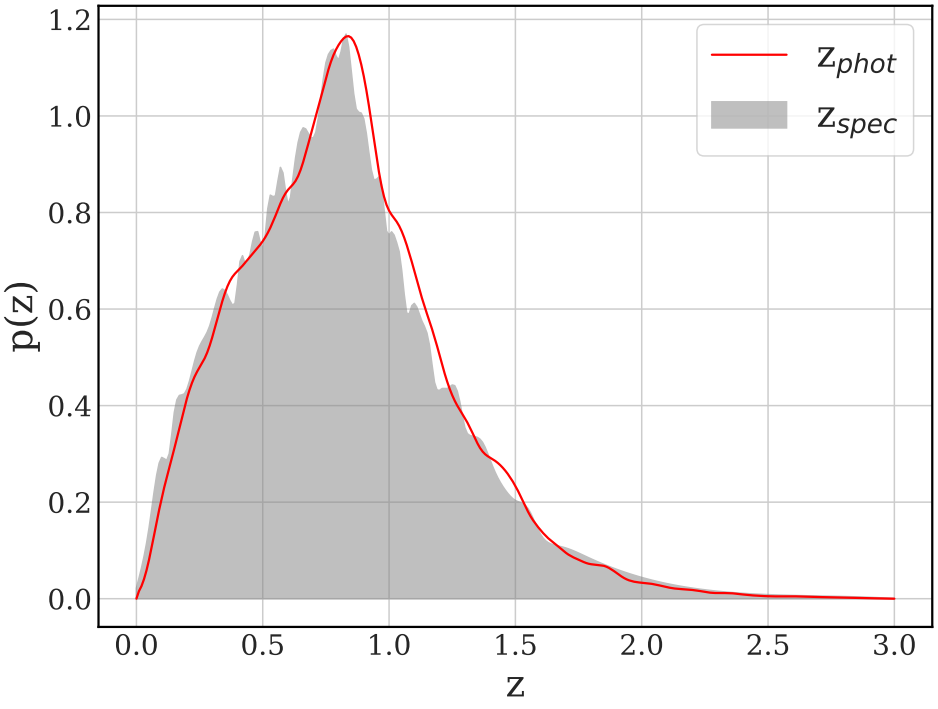


FIGURA 5.11. N(z) para o conjunto de teste.

0.0130, o que indica uma maior ocorrência de predições com probabilidades acumuladas nos extremos da distribuição

Os resultados apresentados mostram que o modelo gerado está razoavelmente calibrado, mas ainda necessita de melhorias para estimar photo-zs de qualidade. Lidar com os outliers e vieses é uma grande questão na estimativa de photo-zs, já que o algoritmo de machine learning se baseia em reconhecer padrões e aqui estamos lidando com relações que podem apresentar padrões semelhantes para significados físicos diferentes, o que impacta diretamente na ciência posterior que se queira fazer com os photo-zs. Para o objetivo do nosso trabalho, que é explorar e entender os processos de ponta a ponta para gerar catálogos de photo-zs, os resultados obtidos são satisfatórios para a próxima etapa.

5.2.3 Distribuição Global de photo-zs e performance do algoritmo

Após executar a etapa 02, como descrito na seção 3.5, temos como resultado as PDFs de photo-zs geradas para o catálogo completo, sem valores de redshift prévios.

A Figura 5.12 mostra o histograma global com os valores de z_{mode} e a PDF global. Vemos que a distribuição gerada pelo modelo é similar ao que vimos para o conjunto de teste, como esperado, uma vez que o conjunto usado para treinar o algoritmo é representativo do conjunto total.

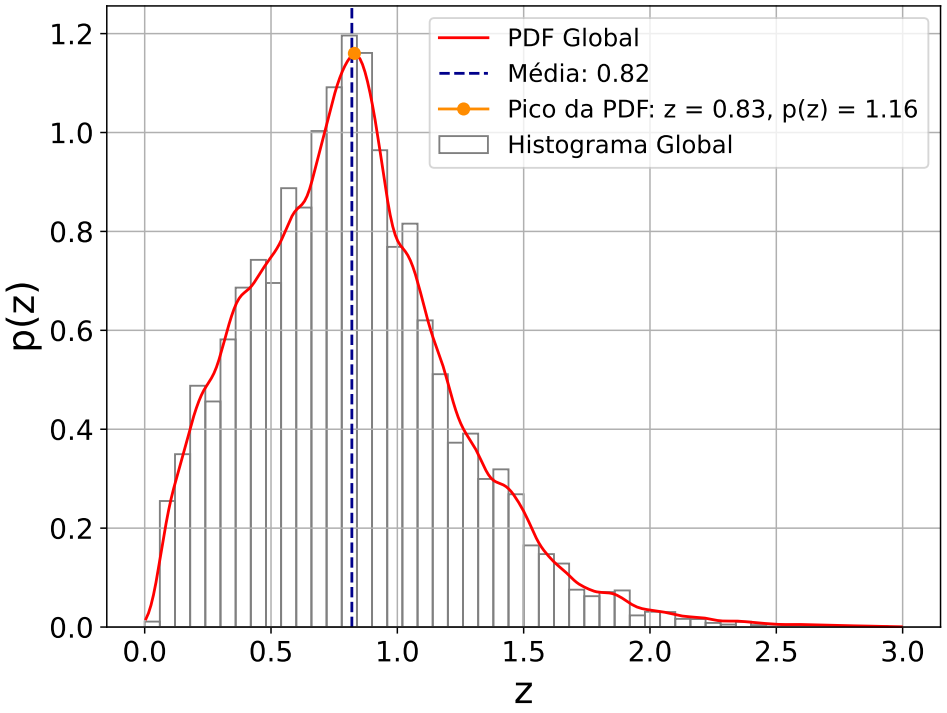


FIGURA 5.12. Distribuição global pdfs.

TABELA 5.2. Performance técnica do TPZ no estágio *Informer*.

Tempo de execução [s]	**Tamanho do modelo [MB]**
55.5	374.6

TABELA 5.3. Performance técnica do TPZ no estágio *Estimate*.

Rodada	**Quantidade de objetos**	**Velocidade [ms/objeto]**	**Tempo de execução [min]**
Etapa 01 (Notebook local)	29912	8.02	4.00
Etapa 02 (Cluster Apollo)	26336000	0.08	37.93

Nas Tabelas 5.2 e 5.3, temos um resumo da performance técnica do algoritmo TPZ nos estágios *Informer* e *Estimate*, respectivamente. Para este último, temos as velocidades de execução na etapa 01, feita em Jupyter Notebook local, e na etapa 02, feita no cluster de computadores Apollo.

O TPZ é um algoritmo com tempo de execução da estimativa maior em relação a outros algoritmos de photo-z. O tamanho do arquivo que armazena o modelo gerado no treino, em formato pickle, também apresenta um tamanho maior. Esse tamanho varia de acordo com a escolha de determinados hiperparâmetros, sendo eles o número de árvores e o valor do *minleaf*. Quanto maior o número de árvores e menor o *minleaf*, maior será o tamanho do modelo.

Na Tabela 5.4, vemos o impacto do *minleaf* no tamanho do modelo. Observamos que, fixando o número de árvores, ao variar o valor de *minleaf* de 2 a 30, o tamanho do arquivo diminuiu em aproximadamente $93,4\%$. Porém, em questão de tempo de execução, a redução do modelo reduziu o tempo de execução em poucos segundos.

Tabela 5.4. Comparação do tamanho do modelos para diferentes *minleaf*

minleaf	**Tamanho do modelo [MB]**	**Tempo de execução [min]**
2	374.6	4.00
5	152.8	3.92
10	75.0	3.90
20	37.2	3.89
30	24.7	3.87

Já na Tabela 5.5, temos a comparação do tempo de execução para dois arquivos de diferentes tamanhos e diferentes números de árvores. Vemos que, apesar do modelo do arquivo 2 ter um tamanho bem menor, seu tempo de execução é bem superior ao modelo do arquivo 1, o que mostra que o número de árvores domina o tempo de execução. Esse é um resultado esperado, pois cada entrada precisa percorrer todas as árvores, respondendo às perguntas de decisão que definem os ramos, até alcançar uma folha onde o valor de redshift é estimado. Então, mesmo que a profundidade de cada árvore seja menor, a quantidade de árvores a ser percorrida influencia mais no custo computacional da predição.

Tabela 5.5. Comparação do tamanho do modelos para diferentes números de árvores.

Número de árvores	**Tamanho do modelo [MB]**	**Tempo de execução [min]**
50	158.5	1.7
300	24.8	12.2

Capítulo 6

Conclusões e Perspectivas

O redshift cosmológico é uma medida de extrema importância para análises cosmológicas, sendo sua obtenção através da fotometria essencial para levantamentos em grande escala, necessitando de medidas robustas para aplicação posterior aos casos científicos. Estimar redshifts fotométricos, por sua vez, é uma tarefa complexa, que envolve lidar com incertezas nos dados e desafios da modelagem estatística.

Nesse contexto, os métodos de machine learning têm se destacado por sua capacidade de lidar com grandes volumes de dados, oferecendo maior flexibilidade na modelagem e bom desempenho, em comparação às outras técnicas.

Um dos aspectos cruciais na estimativa de redshifts fotométricos é o tratamento adequado dos dados de entrada. Como o algoritmos não realizam uma análise científica intrínseca dos padrões que aprendem, cabe ao cientista garantir que os dados fornecidos ao modelo sejam representativos e informativos, minimizando degenerescências nos padrões entre magnitudes e redshifts. Essas degenerescências, que dificultam a distinção entre diferentes valores de redshift a partir de magnitudes similares, são uma das principais fontes de erro nas estimativas. Assim, grande parte do desafio está em encontrar formas de mitigar essas ambiguidades e, consequentemente, reduzir a fração de outliers produzida pelo modelo (Crenshaw et al., 2025).

O TPZ se mostrou um algoritmo capaz de produzir boas estimativas de photo-z, mesmo com algumas melhorias ainda a serem feitas para o conjunto de dados utilizado neste trabalho. As métricas mostraram que o modelo gerado possui uma calibração razoável das PDFs e consegue reproduzir com grande similaridade a distribuição de

galáxias por redshift em comparação à distribuição verdadeira. Observamos que os resultados científicos podem ser sensíveis aos hiperparâmetros, que estão relacionados unicamente com a metodologia do algoritmo. A construção de catálogos a partir do catálogo original, através do hiperparâmetro *nrandom*, introduz pequenas variações nos dados de treinamento, o que ajuda o modelo a generalizar melhor. Com esses catálogos variados, podemos construir árvores de decisão mais profundas (usando *minleaf* pequeno) sem o risco de *overfitting*, porque cada árvore aprende padrões mais robustos em vez de se prender a ruídos específicos. Árvores mais profundas conseguem capturar detalhes importantes, reduzindo a quantidade de outliers causados por degenerescências nos dados. Também vimos que o uso de cores ou magnitudes como atributos não trouxe diferenças significativas para o nosso resultado final, o algoritmo foi capaz de extrair informações relevantes em ambos os casos para o nosso conjunto de treinamento utilizando árvores mais profundas.

No contexto de preparação para os dados reais do LSST, o uso de dados simulados desempenha um papel essencial, permitindo não apenas a testagem dos algoritmos de estimativa de redshifts fotométricos, mas também a validação da infraestrutura computacional necessária. Durante o desenvolvimento deste trabalho, ficou evidente a importância de um ambiente computacional bem estruturado para lidar com grandes volumes de dados, desde a obtenção dos dados até o processamento eficiente das estimativas de photo-z. Trabalhar com simulações realistas permite antecipar os desafios operacionais e metodológicos que surgirão com os dados observacionais, garantindo que tanto os métodos quanto os recursos computacionais estejam preparados para o grande fluxo de informações esperado nos levantamentos pros próximos anos.

Outro aprendizado muito importante durante este trabalho foi a importância da reprodutibilidade para a ciência. Em um projeto de larga escala como o LSST, garantir que análises possam ser verificadas, reproduzidas e estendidas por outros pesquisadores, ou mesmo pelo próprio autor no futuro, é essencial manter uma documentação clara e um controle rigoroso da proveniência dos dados e da metodologia aplicada. O versionamento e a organização de repositórios de código asseguram a rastreabilidade das decisões tomadas ao longo do desenvolvimento, além de possibilitar o acesso estruturado às etapas desenvolvidas. A ausência desse controle pode comprometer a integridade do processo científico, dificultando auditorias, comparações e avanços metodológicos. Assim, práticas como uso de controle de versão, anotações sistemáticas e publicação de

pipelines bem documentados são fundamentais para a robustez das contribuições científicas. O desenvolvimento e aplicação do RAIL e do pipeline PZ Compute também se traduz neste ponto, se mostrando estratégico não apenas por padronizar a avaliação de estimativas de redshifts fotométricos, mas também por facilitar a rastreabilidade e a repetição dos experimentos. Os Jupyter Notebooks escritos para este trabalho estão disponíveis no GitHub, além dos repositórios dos pipelines utilizados, todos sinalizados nos Apêndices B e C. Porém, o acesso aos dados ainda é restrito a usuários com direito aos mesmos. Os conjuntos de dados utilizados neste trabalho poderão ser disponibilizados assim que se tornarem públicos, conforme as diretrizes estabelecidas pela política de dados do LSST.

Mais estudos são necessários, visando aprimorar as análises da estimativa de photo-z. Como perspectivas futuras, temos:

- Comparar o desempenho do TPZ com outros algoritmos de machine learning integrados no RAIL;
- Comparação com o desempenho em dados reais, avaliando os desafios que observações reais adicionam ao problema;
- Comparar resultados com futuras novas implementações do código original do TPZ ao TPZ Lite nos mesmos dados, como a técnica de *out-of-bag* para avaliar os erros no treinamento e medida de importância das variáveis, que permitem identificar quais são os atributos tem maior influência no conjunto de dados de forma mais direta.
- Comparar com outros observáveis - outros tipos de fluxos/magnitudes - e os impactos no resultado final

Em suma, este estudo reforça o potencial dos algoritmos de machine learning na estimação de redshifts fotométricos, ressaltando a importância de uma adequada preparação dos dados, de uma infraestrutura computacional eficiente e da garantia de reprodutibilidade científica para o avanço confiável nessa área.

Referências Bibliográficas

Almosallam, I. A., Jarvis, M. J., & Roberts, S. J. 2016, Monthly Notices of the Royal Astronomical Society, 462, 726–739, doi: 10.1093/mnras/stw1618

Andrew Ng, e. a. 2022, Supervised Machine Learning: Regression and Classification, Coursera. https://www.coursera.org/learn/machine-learning/paidmedia?specialization=machine-learning-introduction

Baum, W. A. 1962, in IAU Symposium, Vol. 15, Problems of Extra-Galactic Research, ed. G. C. McVittie, 390

Carrasco Kind, M., & Brunner, R. J. 2013, MNRAS, 432, 1483, doi: 10.1093/mnras/stt574

Carrasco Kind, M., & Brunner, R. J. 2014, MNRAS, 442, 3380, doi: 10.1093/mnras/stu1098

Cavuoti, S., Amaro, V., Brescia, M., et al. 2017, Monthly Notices of the Royal Astronomical Society, 465, 1959, doi: 10.1093/mnras/stw2930

Chollet, F. 2017, Deep Learning with Python (New York: Manning Publications), ISBN: 9781617294433

Collaboration, D., Abareshi, B., Aguilar, J., et al. 2022, The Astronomical Journal, 164, 207, doi: 10.3847/1538-3881/ac882b

Collister, A., & Lahav, O. 2004, Publications of the Astronomical Society of the Pacific, 116, 345–351, doi: 10.1086/383254

Crenshaw, J. F., Leistedt, B., Graham, M. L., et al. 2025, Quantifying the Impact of LSST u-band Survey Strategy on Photometric Redshift Estimation and the Detection of Lyman-break Galaxies. https://arxiv.org/abs/2503.06016

Crocce, M., Ross, A. J., Sevilla-Noarbe, I., et al. 2019, MNRAS, 482, 2807, doi: 10.1093/mnras/sty2522

Dark Energy Survey Collaboration, Abbott, T., Abdalla, F. B., et al. 2016, MNRAS, 460, 1270, doi: 10.1093/mnras/stw641

de Souza Oliveira Filho, K. 2017, Astronomia e Astrofísica (LF Editorial)

Harrison, E. R. 2020, Cosmology: the science of the universe, 2nd ed (Cambridge University Press)

Ivezić, a. o. 2019, ApJ, 873, 111, doi: 10.3847/1538-4357/ab042c

Izbicki, R., & Lee, A. B. 2017, Electronic Journal of Statistics, 11, 2800 , doi: 10.1214/17-EJS1302

Lambourne, R. J. A. 2010, Relativity, Gravitation and Cosmology (Cambridge University Press)

LIneA (Laboratório Interinstitucional de e-Astronomia). 2025, BRA-LIN-S4.4 – PZ Tables as Federated Datasets, https://linea-it.github.io/pz-lsst-inkind-doc/s4_4/

Loh, E. D., & Spillar, E. J. 1986, ApJ, 303, 154, doi: 10.1086/164062

LSST Dark Energy Science Collaboration (LSST DESC), Abolfathi, B., Alonso, D., et al. 2021, ApJS, 253, 31, doi: 10.3847/1538-4365/abd62c

Margony, B. 1999, Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 357, 93–103, doi: 10.1098/rsta.1999.0316

Marsland, S. 2015, Machine Learning, an algorithmic perspective (CRC Press)

Melchior, P., Moolekamp, F., Jerdee, M., et al. 2018, Astronomy and Computing, 24, 129, doi: 10.1016/j.ascom.2018.07.001

Mellier, Y., Abdurro'uf, Acevedo Barroso, J. A., et al. 2025, Astronomy amp; Astrophysics, 697, A1, doi: 10.1051/0004-6361/202450810

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825

Ryden, B. 2017, Introduction to cosmology (Cambridge University Press)

Schmidt, S. 2025, TPZ: Trees for Photo-Z's, https://github.com/LSSTDESC/rail_tpz/blob/main/examples/TPZ_example_notebook.ipynb

Schmidt, S. J., Malz, A. I., Soo, J. Y. H., et al. 2020, MNRAS, 499, 1587, doi: 10.1093/mnras/staa2799

Soo, J. Y. H. 2018, PhD thesis, University College London

Team, T. R., van den Busch, J. L., Charles, E., et al. 2025, Redshift Assessment Infrastructure Layers (RAIL): Rubin-era photometric redshift stress-testing and at-scale production. https://arxiv.org/abs/2505.02928

Waga, I. 2005, Revista Brasileira de Ensino de Física. https://www.scielo.br/j/rbef/a/mHkGCJLdDGnhQgGwTQfxxGj/?lang=pt

Zěljko Ivezić, Andrew J. Connolly, J. T. V., & Gray, A. 2020, Statistics, data mining, and machine learning in astronomy : a practical Python guide for the analysis of survey data (Princeton University Press)

Apêndice A

Apêndice A - Métrica FRW e a relação do redshift cosmológico com o fator de escala

A métrica de um espaço é uma descrição geométrica de como calcular a distância entre dois pontos nesse espaço. Para a construção da métrica utilizada na descrição do Universo, o ponto de partida aqui será as equações de Lorentz para relatividade restrita.

A.1 Equações de Lorentz

Tomando dois referenciais inerciais S e S' e simplificando inicialmente o movimento apenas na direção x, pode-se definir as funções f e g que mapeiam $(x,t) \to (x',t')$, tal que

$$x' = f(x,t) \text{ e } t' = g(x,t). \tag{A.1}$$

Como se tratam e referenciais inerciais, tem-se que uma partícula se movendo livremente tem velocidade constante. Sendo assim, o gráfico x x t é uma linha reta, então a transformação $(x,t) \to (x',t')$ é linear, permitindo escrever

$$x' = \alpha_1 x + \alpha_2 t \text{ e } t' = \alpha_3 x + \alpha_4 t, \tag{A.2}$$

onde os coeficientes $\alpha_n,\ n = 1, 2, 3, 4$ dependem de v, para manter a homogeneidade e isotropia do espaço.

Uma vez que S' tem velocidade v relativa a S, um observador situado em $x' = 0$ se move com trajetória $x = vt$ em S. Com isso, tem-se que

$$0 = \alpha_1 vt + \alpha_2 t \tag{A.3}$$

$$\alpha_2 = -\alpha_1 v. \tag{A.4}$$

Substituindo (26) em (24) para x'

$$x' = \alpha_1 (x - vt). \tag{A.5}$$

Analogamente para $-v$, tem-se como resultado

$$x' = \alpha_1 (x' + vt'). \tag{A.6}$$

É importante observar que $\alpha_v = \alpha_{-v}$. Para determinar o valor de α_1, será utilizado o segundo postulado da relatividade restrita, que diz que a velocidade da luz é igual em ambos os referenciais. Sendo assim,

$$x = ct \text{ e } x' = ct', \tag{A.7}$$

que pode ser reescrito como

$$ct = \alpha_1 (ct' + vt') \text{ e } ct' = \alpha_1 (ct - vt). \tag{A.8}$$

Isolando α_1, tem-se que

$$\alpha_1^2 = \frac{ct}{ct' - vt'} \cdot \frac{ct'}{ct - vt} = \frac{1}{1 - \frac{v^2}{c^2}} \tag{A.9}$$

Chamando, finalmente, α_1 de γ, chega-se à expressão

$$\gamma = \sqrt{\frac{1}{1 - \frac{v^2}{c^2}}}, \tag{A.10}$$

mostrando que para velocidades muito menores que a da luz - $v << c$ - as transformações de Lorentz se aproximam das transformações de Galileu e que para $v > c$ tem-se uma raiz complexa, indicando um resultado incompatível. Agora, podemos construir a transformação para o tempo. Substituindo (27) em (28)

$$x = \gamma(\gamma(x - vt) + vt') \tag{A.11}$$

$$t' = \frac{x - \gamma^2 x + \gamma^2 vt}{\gamma v} \tag{A.12}$$

Com a expressão (32), temos que

$$t' = \gamma(t - \frac{v}{c^2}x) \tag{A.13}$$

Novamente, para $\frac{v}{c} << 1$, nos aproximamos das transformações de Galileu, onde $t \approx t'$.

Podemos estender as transformações para os eixos y e z, de forma que, dadas as condições descritas inicialmente e sendo y e z perpendiculares à direção do movimento, temos

$$y' = k_1 y \text{ e } z' = k_2 z. \tag{A.14}$$

Por simetria, $y = k_1 y'$ e $z = k_2 z'$. Dessa forma, k_1 e k_2 podem ter valores 1 ou -1. Como para -1 não temos a transformação identidade em $v = 0$, então

$$k_1 = k_2 = 1 \tag{A.15}$$

e, então,

$$y = y' \text{ e } z = z' \tag{A.16}$$

Por fim, (27), (35) e (38) representam as transformações de Lorentz para a relatividade restrita.

A.2 Espaço-tempo de Minkowski

É importante determinar qual é o intervalo invariante no espaço que estamos tratando, que nada mais é do que a separação entre dois eventos, pois esta é uma propriedade intrínseca dos mesmos, ou seja, independe do referencial. Sendo assim, podemos analisar eventos ocorridos em outro referencial usando o intervalo invariante no nosso referencial.

Em um espaço tridimensional, representamos a distância entre dois pontos por

$$d^2 = \Delta x^2 + \Delta y^2 + \Delta z^2 \tag{A.17}$$

Se um sinal se propaga na velocidade da luz, temos então que

$$d^2 = c^2 \Delta t^2 \tag{A.18}$$

Logo, podemos escrever

$$-c^2 \Delta t^2 + \Delta x^2 + \Delta y^2 + \Delta z^2 = 0 \tag{A.19}$$

A expressão acima mostra a condição para dois eventos separados por um feixe de luz. Devido ao segundo postulado da relatividade restrita, se esse intervalo entre dois eventos é zero em um referencial, ele será zero em qualquer referencial, o que nos dá a quantidade invariante que buscamos. Ela pode ser generalizada para qualquer intervalo que separe dois eventos no espaço em questão, de modo que o intervalo invariante é descrito por

$$\Delta s^2 = -c^2\Delta t^2 + \Delta x^2 + \Delta y^2 + \Delta z^2 = -c^2\Delta t^2 + \Delta l^2, \tag{A.20}$$

definindo, assim, a métrica do espaço-tempo de Minkowski, onde temos a adição do termo relacionado ao tempo em comparação ao espaço euclidiano.

A.3 Geometrias

Uma vez que, a partir do princípio cosmológico, assume-se que o universo é homogêneo e isotrópico para grandes escalas, é necessário descrevê-lo em geometrias que apresentem essas propriedades. Há três espaços que podem ser utilizados para tal:

1. Espaço plano - Euclidiano

O espaço mais simples que atende às propriedades buscadas é o espaço plano, denotado por $\mathbf{R}^3$. Definindo ds como o elemento de distância infinitesimal, tem-se que neste espaço

$$ds^2 = dx^2 + dy^2 + dz^2. \tag{A.21}$$

Que, escrito em coordenadas esféricas, se torna

$$ds^2 = dr^2 + r^2(d\theta^2 + sen^2 d\phi). \tag{A.22}$$

2. Curvatura positiva - Esfera

Como próxima geometria, temos uma esfera 3D de raio R construída em um espaço Euclidiano quadri-dimensional $\mathbf{R}^4$, descrita por

$$x^2 + y^2 + z^2 + w^2 = R^2 \tag{A.23}$$

Novamente, buscando escrever ds em coordenadas esféricas, pode-se escrever $w = r^2 - R^2$, de forma que

$$dw = -\frac{rdr}{\sqrt{R^2 - r^2}} dr^2. \tag{A.24}$$

Sendo assim, o elemento ds é dado por

$$ds^2 = \frac{R^2}{R^2 - r^2} dr^2 + r^2(d\theta^2 + sen^2 d\phi). \tag{A.25}$$

3. Curvatura negativa - Hipérbole

Por fim, tem-se como último espaço uma hipérbole 3D construída em $\mathbf{R}^4$, dada por

$$x^2 + y^2 + z^2 - w^2 = -R^2 \tag{A.26}$$

Desta forma, agora tem-se, analogamente ao caso anterior, que ds pode ser escrito como

$$ds^2 = \frac{R^2}{R^2 + r^2} dr^2 + r^2(d\theta^2 + sen^2 d\phi). \tag{A.27}$$

Pode-se então estabelecer uma métrica geral, adotando um k tal que

$$ds^2 = \frac{1}{1 - k\frac{r^2}{R^2}} dr^2 + r^2(d\theta^2 + sen^2 d\phi), \text{com } k = \begin{cases} +1 & \text{(Esférico)} \\ 0 & \text{(Euclidiano)} \\ -1 & \text{(Hiperbólico)} \end{cases} \tag{A.28}$$

A.4 Métrica FRW

Como vimos anteriormente, a equação (42) do espaço-tempo de Minkowski tem componente espacial plana. Por isso, descrevemos anteriormente uma geometria que considerasse o espaço que queremos descrever - o Universo em expansão.

Além dessa descrição geométrica, é necessário também adicionar à componente espacial o fator de escala, como descrito na seção 3.1.2. Subsitutindo, então, a componente espacial da métrica de Minkowski, temos

$$ds^2 = -c^2dt^2 + a(t)^2[\frac{1}{1-k\frac{r^2}{R^2}}dr^2 + r^2(d\theta^2 + sen^2d\phi)], \tag{A.29}$$

definida como a métrica de Friedmann-Robertson-Walker.

A.5 Redshift e fator de escala

Conforme as condições descritas na seção 3.1.2, podemos rescrever a métrica de forma que

$$\frac{cdt}{a(t)} = \frac{dr}{\sqrt{1-\frac{kr^2}{R^2}}} \tag{A.30}$$

Integrando ambos os lados:

$$c\int_{t_1}^{t_o}\frac{dt}{a(t)} = \int_0^{r_1}\frac{dr}{\sqrt{1-\frac{kr^2}{R^2}}} \tag{A.31}$$

Para um pico sucessivo emitido em $t_1 + \delta t_1$ e observado em $t_o + \delta t_o$:

$$c\int_{t_1+\delta t_1}^{t_o+\delta t_o}\frac{dt}{a(t)} = \int_0^{r_1}\frac{dr}{\sqrt{1-\frac{kr^2}{R^2}}} \tag{A.32}$$

Unindo os dois resultados, temos

$$c\int_{t_1+\delta t_1}^{t_o+\delta t_o}\frac{dt}{a(t)} = c\int_{t_1}^{t_o}\frac{dt}{a(t)} \tag{A.33}$$

Como a variação de tempo da integral é muito menor que a escala de tempo de Universo, podemos tomar o fator de escala constante no instante em questão. Logo

$$\frac{1}{a(t_1)}\int_{t_1+\delta t_1}^{t_o+\delta t_o}dt = \frac{1}{a(t_o)}\int_{t_1}^{t_o}dt \tag{A.34}$$

Utilizando as propriedades de integrais:

$$\int_{t_1+\delta t_1}^{t_o} dt \; - \int_{t_1+\delta t_1}^{t_o} dt \; + \int_{t_1+\delta t_1}^{t_o+\delta t_o} dt \; = \int_{t_1}^{t_o} dt \tag{A.35}$$

$$\int_{t_o}^{t_1+\delta t_1} dt \; + \int_{t_1+\delta t_1}^{t_o+\delta t_o} dt \; = \int_{t_1}^{t_o} dt \; + \int_{t_o}^{t_1+\delta t_1} dt \tag{A.36}$$

$$\int_{t_o}^{t_o+\delta t_o} dt \; = \int_{t_1}^{t_1+\delta t_1} dt \tag{A.37}$$

Voltando para a expressão original:

$$\frac{1}{a(t_o)} \int_{t_o}^{t_o+\delta t_o} dt \; = \frac{1}{a(t_1} \int_{t_1}^{t_1+\delta t_1} dt \tag{A.38}$$

$$\frac{1}{a(t_1)}[t_1 + \delta t_1 - t_1] = \frac{1}{a(t_o)}[t_o + \delta t_o - t_o] \tag{A.39}$$

$$\frac{\delta t_o}{a(t_o)} = \frac{\delta t_1}{a(t_1)} \tag{A.40}$$

Uma vez que δt expressa o período da onda de luz, podemos escrever

$$\delta t = \frac{1}{f} = \frac{\lambda}{c} \tag{A.41}$$

Então, para o nosso caso

$$\delta t_o = \frac{\lambda_o}{c} \tag{A.42}$$

$$\delta t_1 = \frac{\lambda_1}{c} \tag{A.43}$$

Substituindo no resultado das integrais:

$$\frac{\lambda t_o}{a(t_o)c} = \frac{\lambda t_1}{a(t_1)c} \tag{A.44}$$

$$\lambda t_o = \lambda t_1 \frac{a(t_o)}{a(t_1)} \tag{A.45}$$

Temos, então, o resultado comentado inicialmente, que o comprimento de onda do sinal luminoso que chega até o observador é aumentado pela expansão do universo.

O redshift é definido quantitativamente por

$$z = \frac{\lambda_o - \lambda_1}{\lambda_1} \tag{A.46}$$

Substituindo no resultado anterior, temos:

$$z = \frac{\lambda_o - \lambda_o \frac{a(t_1)}{a(t_o)}}{\lambda_o \frac{a(t_1)}{a(t_o)}} \tag{A.47}$$

$$1 + z = \frac{a(t_o)}{a(t_1)} \tag{A.48}$$

Apêndice B

Apêndice B - Etapa 01 da estimativa do photo-z

Os notebooks e scripts utilizados neste trabalho estão em sua maioria em repositórios no GitHub. Neste apêndice, vamos descrever como cada etapa foi feita computacionalmente, apontando para as ferramentas utilizadas.

B.1 Obtenção dos dados

Utilizamos o notebook **1_Data_Preparation.ipynb**, disponível em `https://github.com/linea-it/pz-lsst-inkind/blob/main/old/dp02_data_preparation`, para gerar a *skinny table*, através da plataforma OnDemand, onde é possível rodar os notebooks no cluster de computadores. Seguimos as instruções sinalizadas no arquivo **README.md** do mesmo diretório para criar o ambiente e o kernel para o notebook.

Configuramos o arquivo **config_lsst_dp02.yaml** para habilitar a conversão das magnitudes em fluxos, filtrar em detect_isPrimary, fazer correção de avermelhamento, selecionar as colunas de interesse e tratar os valores inválidos como NaNs. O arquivo com todas as configurações usadas está disponível em `https://github.com/andreiadourado/tcc_photoz/tree/main/pz-lsst-inkind-files/quality_cuts`.

B.2 Cortes de qualidade

Os cortes de qualidade na *skinny table* foram aplicados utilizando o script **utils/-post_pre_processing_cleaning.py**, adaptado para os cortes que aplicamos. O arquivo adaptado está disponivel em `https://github.com/andreiadourado/tcc_photoz/blob/main/scripts_pz_lsst_inkind/quality_cuts/post_pre_processing_cleaning.py`.

B.3 Seleção dos objetos do *training set*

Para selecionar a fração aleatória dos objetos da *skinny table*, utilizamos o script **DP02_step_1_random_object_catalog.py**, disponível em `https://github.com/linea-it/pz-lsst-inkind/blob/main/old/random_samples_scripts`, adaptado para fazer apenas a seleção aleatória. O arquivo adaptado está disponivel em `https://github.com/andreiadourado/tcc_photoz/blob/main/scripts_pz_lsst_inkind/random_catalog/random_catalog.py`.

B.4 Crossmatching

B.4.1 Biblioteca LSDB

A LSDB é uma biblioteca que permite processar grandes volumes de dados astronômicos de forma escalável, utilizando o framework Dask. As operações, como consulta e cruzamento de dados no formato HATS, são organizadas em um grafo de tarefas e executadas paralelamente por múltiplos workers distribuídos em um cluster, otimizando desempenho e eficiência computacional, o que é ideal no cenário do volume de dados previsto para o LSST. Mais informações sobre a biblioteca podem ser lidas em `https://docs.lsdb.io/en/stable/`.

O formato HATS faz a partição de objetos em uma esfera. Ele utiliza a pixelização HEALPix, adaptando o tamanho dos pixels conforme a densidade de objetos no céu, para equilibrar a carga por partição. A estrutura de arquivos segue uma hierarquia organizada, com metadados e dados armazenados em diretórios baseados

na ordenação e posição dos pixels esféricos. A conversão dos catálogos para o formato HATS é feita através da ferramenta HATS Import. As documentações de ambos podem ser acessadas em `https://hats.readthedocs.io/en/stable/` e `https://hats-import.readthedocs.io/en/stable/`.

B.4.2 Notebooks

Para fazer o crossmatching, utilizamos os notebooks disponíveis em `https://github.com/linea-it/pz-lsst-inkind/tree/main/old/make_hats_notebooks`. O notebook **cleaning_dp02_truth_match.ipynb** faz a reorganização dos dados da tabela Truth, o notebook **make_collection.ipynb** cria os arquivos no formato HATS e o notebook **make_xmatch.ipynb** utiliza esses arquivos para fazer o crossmatching. Os respectivos arquivos de configuração para cada notebook adaptados estão disponíveis em `https://github.com/andreiadourado/tcc_photoz/tree/main/scripts_pz_lsst_inkind/crossmatching`.

Esses scripts originam o pipeline Training Set Maker, desenvolvido para criar conjuntos de treinamento para algoritmos de photo-z baseados em ML, possibilitando a obtenção dos spec-zs. Mais detalhes sobre podem ser lidos em `https://linea-it.github.io/pz-lsst-inkind-doc/s4_1/`.

O pipeline estará disponível na plataforma PZ-Server, que será melhor detalhada em B.2.

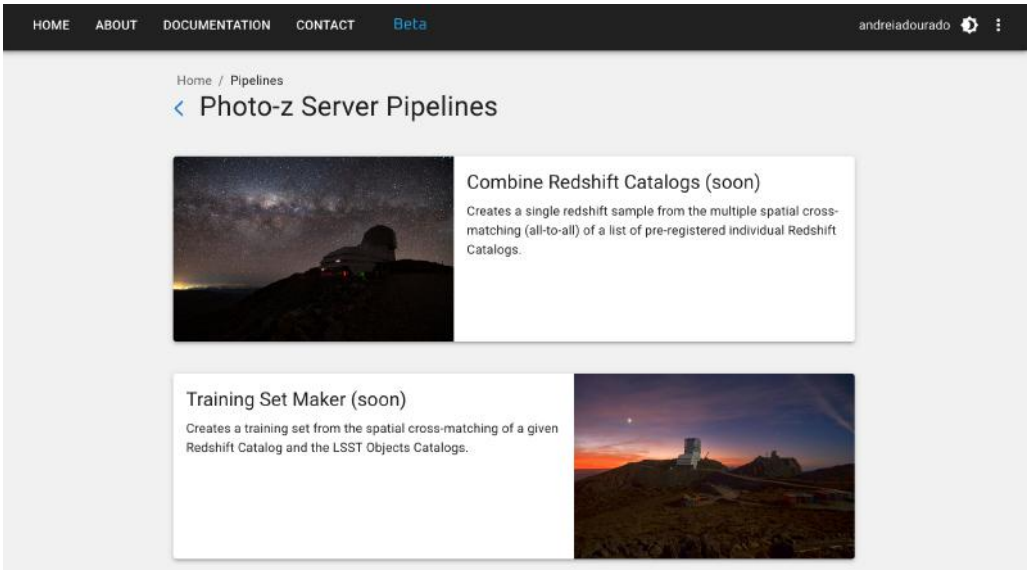


FIGURA B.1. Pipelines que serão disponibilizados no PZ-Server.]

B.5 Treino e Teste do algoritmo

Os notebooks descritos nessa seção estão disponíveis em `https://github.com/andreiadourado/tcc_photoz/notebooks`.

As figuras de caracterização do *training set* mostradas no capítulo 4, foram geradas com o notebook **01_QA_training_set.ipynb**; as divisões dos conjuntos de treino e teste foram feitas no notebook **02_files_to_run.ipynb**.

No notebook **03_run_tpz_lite.ipynb**, temos as etapas de treino e teste para o TPZ utilizando o RAIL, e no notebook **04_metrics.ipynb** foram geradas as figuras de métricas da seção 5.2.

B.6 Photo-z Server

Também como parte da contribuição in-kind, o LIneA desenvolveu uma plataforma online para hospedar dados leves de photo-zs, que permite o compartilhamento de produtos de dados entre os usuários autorizados, atualmente operando em sua versão Beta. Possui um repositório público que pode ser acessado em `https://github.com/linea-it/pzserver_app`, e mais informações podem ser lidas em `https://linea-it.github.io/pz-lsst-inkind-doc/s4_2/`.

Uma imagem da página inicial do pz-server pode ser vista na figura B.2

Os produtos de dados dos usuários podem ser acessados na aba *User-generated Data Products*. Um produto de dado pode ser adicionado no botão "New product", como ilustrado na figura B.3, que leva para a página (figura B.4) onde as descrições, tipo de produto e arquivos principais e auxiliares serão adicionados para fazer o upload.

A seguir, descreveremos os tipos de produtos gerados neste trabalho e os arquivos que cada um contém, além de figuras de exemplos das páginas dos produtos na plataforma:

- *Objetc catalogs*: skinny tables geradas e versões html dos notebooks de caracterização das mesmas;

- *Training set*: arquivo de formato parquet com o training set e versão html do notebook de caracterização **01_QA_training_set.ipynb** (figura B.5);
- *Training results*: arquivo de formato pickle com o modelo gerado;
- *Validation Results*: arquivo de formato hdf5 e versão html do notebook **04_metrics.ipynb** com as métricas do conjunto de teste (figura B.6);
- *Photo-z Estimates*: tabela de photo-zs e versão html do notebook com os resultados do catálogo final e descrição para acesso aos dados.

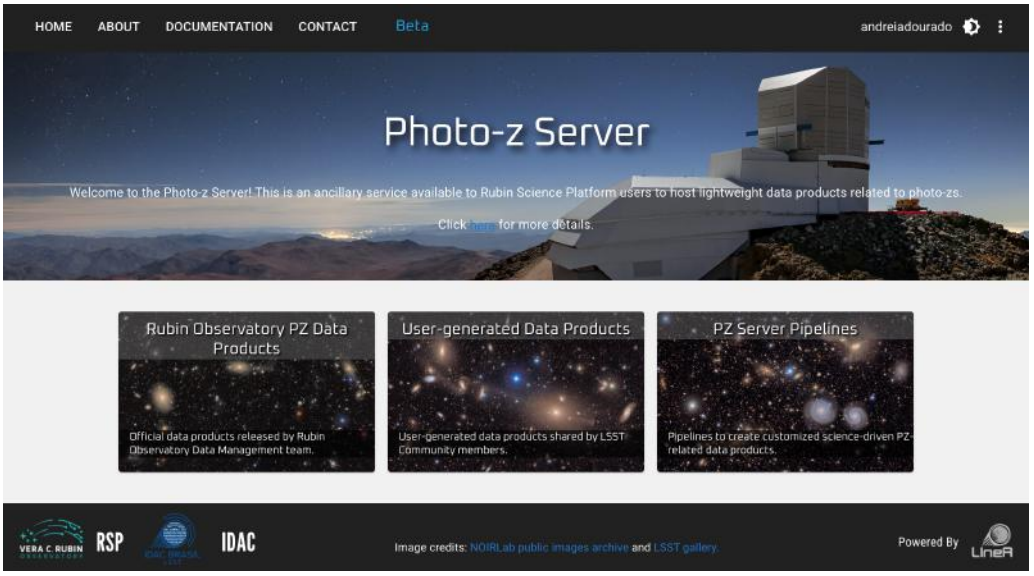


FIGURA B.2. Página inicial do Pz-Server

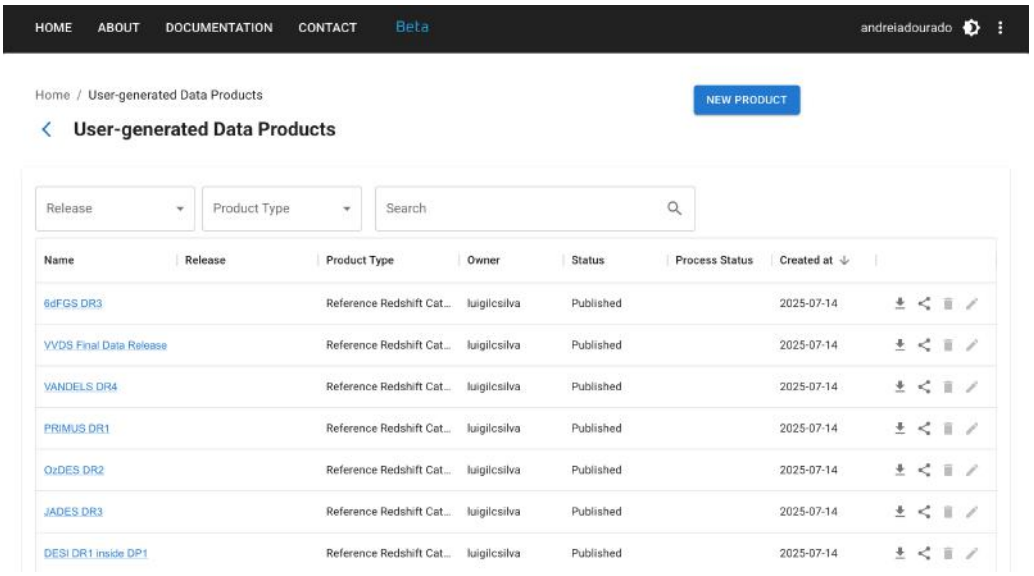


FIGURA B.3. Página inicial da aba *User-generated Data Products*

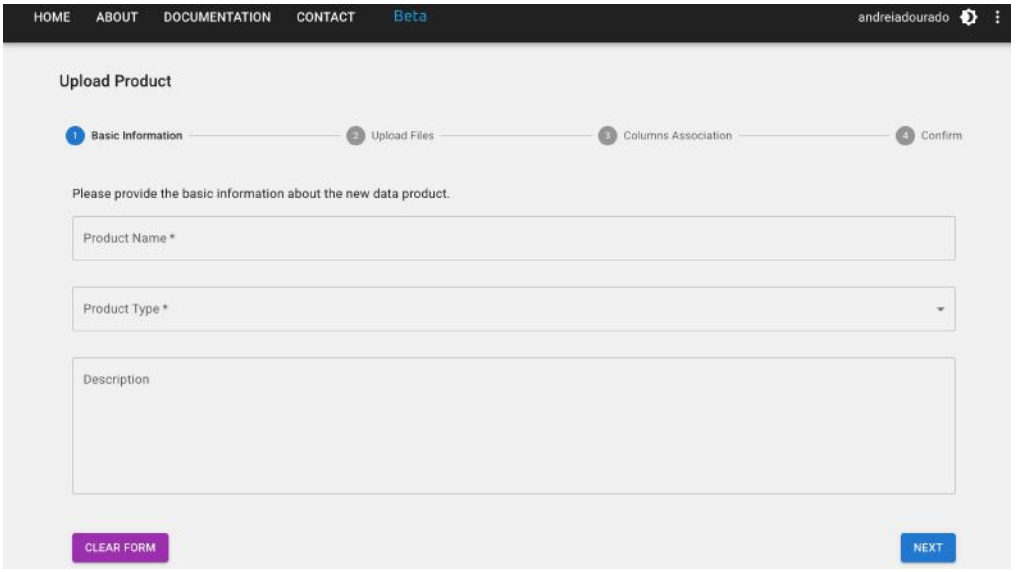


FIGURA B.4. Página inicial para upload de um novo produto de dado.

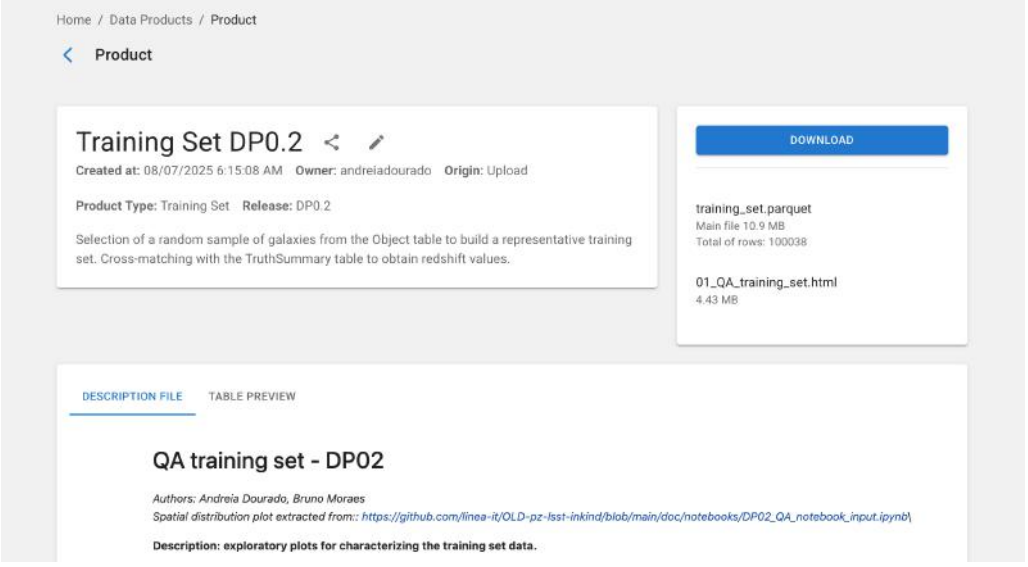


FIGURA B.5. Página do produto de dado *Training set*

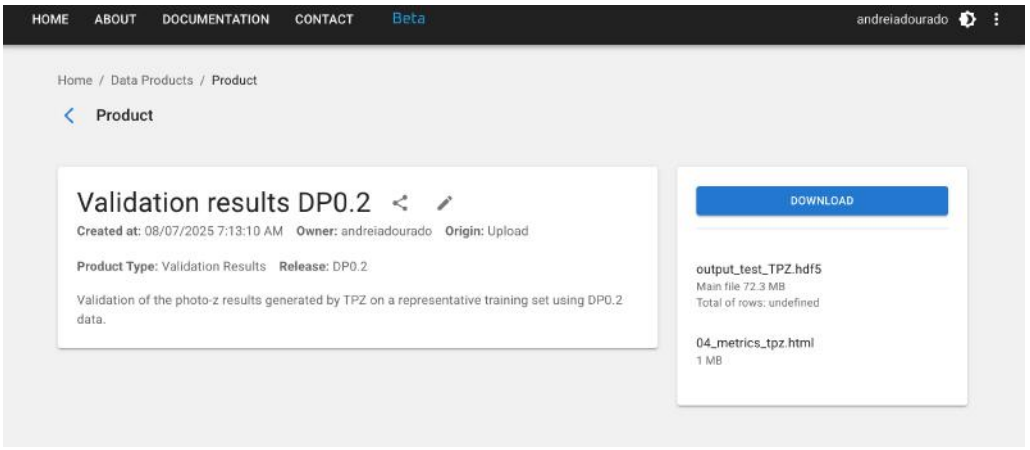


FIGURA B.6. Página do produto de dado *Validation results*

Apêndice C

Apêndice C - Etapa 02 da estimativa do photo-z

C.1 Cluster Apollo

O Cluster Apollo - HPE Apollo 2000 é uma infraestrutura de computação de alto desempenho com 28 nós e 1072 cores físicos, totalizando 2144 cores disponíveis pela aplicação do Hyper-Threading. Possui áreas distintas de armazenamento: uma área *scratch* no sistema Lustre, para arquivos temporários durante a execução de *jobs*, e uma *home* acessível apenas via nó de login. A submissão de jobs é feita por meio de scripts com diretivas do sistema de gerenciamento Slurm, e o uso do cluster é organizado por partições e contas que determinam prioridades. Mais informações pode ser encontradas em `https://docs.linea.org.br/processamento/apollo/index.html`.

C.2 Configurações do pz-compute

No diretório criado para execução do pz-compute, temos os diretórios de input, output e logs, o arquivo **estimator_tpz.pkl** com o modelo gerado e os arquivos de configuração. O arquivo **tpz.yaml** define as configurações do algortimo e o arquivo **pz-compute.yaml** define as configurações do pipeline.

Para o algoritmo, como executamos a etapa de teste e o modelo carrega as configurações definidas no treino, precisamos definir no arquivo de configuração apenas os

nomes das colunas de magnitude e erros. Nas configurações do pipeline, definimos o nome do algoritmo, o nome do arquivo que contém o modelo, o arquivo de configuração do algoritmo e os parâmetros de configuração do cluster. Utilizamos 9 nós (parâmetro -N), sendo eles o nó 18 e os nós de 20 a 27 (parâmetro -w); 90 tarefas totais disrtribuídas entre os nós (parâmetro -n), com 10 CPUs alocadas para cada tarefa (parâmetro -c). A figura C.1 mostra os parâmetros definidos no arquivo. Os arquivos de configuração usados podem ser acessados em `https://github.com/andreiadourado/tcc_photoz/tree/main/pz-compute-config`.

Figura C.1.
Arquivo do configuração do pipeline pz-compute.

C.3 Post-processing

Para fazer a leitura e validação do output da rodada para a *skinny table*, que gerou a figura de N(z) global, utilizamos a etapa de post-processing do notebook . A versão adaptada foi feita no notebook **05_photoz_validation.ipynb** e está disponível em `https://github.com/andreiadourado/tcc_photoz/tree/main/notebooks`.