CEFET/RJ - CENTRO FEDERAL DE EDUCACAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA

Técnicas de Aprendizado de Máquina para Predição de Desvio Fotométrico

Jorge Gabriel Proença de Almeida Rodrigues

Cássio Fernando Brito de Souza

Orientador: Eduardo Bezerra, D.Sc.

Co-Orientador: Ricardo Ogando, D.Sc.

Rio de Janeiro

Março de 2022

CEFET/RJ - CENTRO FEDERAL DE EDUCACAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA

Técnicas de Aprendizado de Máquina para Predição de Desvio Fotométrico

Jorge Gabriel Proença de Almeida Rodrigues

Cássio Fernando Brito de Souza

Projeto final apresentado em cumprimento às normas do Departamento de Educação Superior do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Eduardo Bezerra, D.Sc.

Co-Orientador: Ricardo Ogando, D.Sc.

Rio de Janeiro,

Março de 2022

RESUMO

Com o advento da captura de grandes quantidades de dados na Astronomia, algoritmos de aprendizado de máquina têm se tornado cada vez mais comum nos processos de análise e predição nessa área do conhecimento. Uma das aplicações desses algoritmos é na análise, validação e predição do desvio fotométrico. Em conjuntos de dados correspondentes a essa aplicação, são registradas não apenas as magnitudes do objeto medidas em diferentes bandas do espectro eletromagnético, mas também suas respectivas medidas de erro. Nesse contexto, Fialho [2020] buscou investigar a relevância da utilização dos erros durante o treinamento dos modelos de aprendizado de máquina para predição do desvio fotométrico. Uma limitação desse trabalho é que apenas uma estratégia de mapeamento entre magnitudes e erros foi investigada. Este presente trabalho tem como objetivo investigar diferentes estratégias para mapear valores de magnitude para erros durante o treinamento de um modelo para predição de desvio fotométrico. Para cada estratégia proposta, experimentos computacionais são realizados com objetivo de comparar e avaliar a sua efetividade na tarefa de predição.

Palavras-chaves: Aprendizado de Máquina; Redes Neurais Artificiais; Desvio Fotométrico

ABSTRACT

With the advent of capturing large amounts of data in Astronomy, machine learning algorithms have become increasingly common in the analysis and prediction processes in this area of knowledge. One of the applications of these algorithms is in the analysis, validation and prediction of photometric deviation. In data sets corresponding to this application, not only the object magnitudes measured in different bands of the electromagnetic spectrum are recorded, but also their respective error measures. In this context, Fialho [2020] sought to investigate the relevance of using errors during the training of machine learning models to predict photometric deviation. A limitation of this work is that only one mapping strategy between magnitudes and errors was investigated. This present work aims to investigate different strategies to map magnitude values to errors during the training of a model for photometric deviation prediction. For each proposed strategy, computational experiments are performed in order to compare and evaluate its effectiveness in the prediction task.

Keywords: Machine Learning; Artificial Neural Networks; Photometric Redshift

Sumário

Lista de Figuras

Lista de Tabelas

LISTA DE ABREVIAÇÕES

COIN	Programa *Cosmostatistics Initiative*	5, 26
DT-M-X-M	Regressor Decision Tree Com Estratégia Muitos Para Muitos	41
DT-M-X-1	Regressor Decision Tree Com Estratégia Muitos Para Um	41
DT-1-X-1	Regressor Decision Tree Com Estratégia Um Para Um	41
HAPPY	*COIN/HAPPY*	5, 25, 26, 33, 39, 43, 47
IR-1-X-1	Regressor Isotonic Com Estratégia Um Para Um	41
KNN-M-X-M	Regressor K Nearest Neighbors Com Estratégia Muitos Para Muitos	41
KNN-M-X-1	Regressor K Nearest Neighbors Com Estratégia Muitos Para Um	41
KNN-1-X-1	Regressor K Nearest Neighbors Com Estratégia Um Para Um	41
MLP-M-X-M	Regressor Multilayer Perceptron Com Estratégia Muitos Para Muitos	41
MLP-M-X-1	Regressor Multilayer Perceptron Com Estratégia Muitos Para Um	41
MLP-1-X-1	Regressor Multilayer Perceptron Com Estratégia Um Para Um	41
MSE	*Mean Squared Error*	5, 42, 43
RF-M-X-M	Regressor Random Forest Com Estratégia Muitos Para Muitos	41
RF-M-X-1	Regressor Random Forest Com Estratégia Muitos Para Um	41
RF-1-X-1	Regressor Random Forest Com Estratégia Um Para Um	41
SDSS	*Sloan Digital Sky Survey*	1, 5, 10, 11, 12, 18, 47, 48
SKLEARN	*Scikit-Learn*	42
TEDDY	*COIN/TEDDY*	5, 25, 26, 33, 39, 43, 47
XGB-M-X-M	Regressor XGBoost Com Estratégia Muitos Para Muitos	41, 43
XGB-M-X-1	Regressor XGBoost Com Estratégia Muitos Para Um	41
XGB-1-X-1	Regressor XGBoost Com Estratégia Um Para Um	41

Capítulo 1

Introdução

1.1 Contextualização

O efeito Doppler é um fenômeno observado quando um objeto emissor de ondas está em movimento em relação a um observador. No caso da aproximação, há uma diminuição no tamanho do comprimento das ondas (caracterizando o *blueshift*). No caso de afastamento, há um aumento no tamanho do comprimento das ondas (caracterizando o *redshift*). Uma observação comum desse fenômeno pode ser realizada ao ouvir uma sirene (seja do caminhão dos bombeiros, da ambulância ou de um carro policial) em movimento. É possível notar que na aproximação, o tom das sirenes é mais agudo. E no afastamento, o tom das sirenes é mais grave [Kaufmann and Comins, 2014].

No contexto desse trabalho, o efeito Doppler é observado em objetos distantes (galáxias, quasares e estrelas). Nesse caso, o astrônomo Edwin Hubble [1929] percebeu que quanto mais distante um objeto, mais rápido ele se afasta devido a expansão espacial do universo. O desvio fotométrico é justamente o *redshift* observado por causa dessa expansão [Kaufmann and Comins, 2014].

O desvio fotométrico é de suma importância para a Astronomia, pois além de comprovar a observação feita por Edwin Hubble, a partir dele podemos obter propriedades e fatos fundamentais para entender o universo. Uma dessas propriedades é a constante de Hubble que indica o quão rápido o universo se expande. A partir dela também se obtém a idade do universo e uma das bases para a teoria do Big Bang [Kaufmann and Comins, 2014, pag. 531].

Por esses motivos, nas últimas décadas, os equipamentos utilizados para realizar observações astronômicas vêm levantado uma quantidade enorme de dados. Devido ao tamanho desses dados, tornam-se inviáveis serem realizadas tarefas como classificação e regressão dessas observações pelos seres humanos. Exemplos desses equipamentos são os telescópios utilizados no *Sloan Digital Sky Survey* (SDSS) que podem gerar 200 bilhões de gigabytes de dados observacionais em uma noite [Kremer et al., 2017].

Para lidar com essas quantidades de dados coletadas nos levantamentos astronômicos, algo-

ritmos de Aprendizado de Máquina têm sido cada vez mais utilizados em levantamentos de dados astronômicos para resolver problemas como classificação de galáxias ou predição de desvio fotométrico. Os algoritmos de Aprendizado de Máquina possuem duas vertentes: aprendizado supervisionado e não supervisionado. No aprendizado supervisionado, existe uma variável alvo (*target*), uma variável a ser modelada por um processo preditivo. Já no caso do aprendizado não supervisionado, o algoritmo de aprendizado deve se encarregar de encontrar os padrões nos dados de treinamento sem a existência de uma variável alvo [Mitchell, 1997]. Ao usar algoritmos de aprendizado supervisionado, modelos são alimentados com uma grande quantidade de exemplos com o propósito de ajustar seus parâmetros, procedimento conhecido como *treinamento* ou *aprendizado*. Após a realização do procedimento de ajuste de seus parâmetros, é esperado que esses modelos adquiram a capacidade de induzir resultados com boa precisão.

O problema da predição de desvio fotométrico pode ser enquadrado como uma tarefa de aprendizado supervisionado. Nesse caso, a variável alvo é o próprio desvio fotométrico. Visto que essa variável é contínua, estamos diante de um problema de regressão.

Na Astronomia existe o termo chamado *magnitude*, que é o brilho aparente de um objeto (galáxia ou quasar) variando a uma certa distância de comprimento de onda (mais detalhes na Seção 2.1.4). A magnitude de um objeto está relacionada com a distância desse objeto. A escala usada para medir magnitudes é invertida, no sentido de que, quanto mais próximo o objeto está do ponto de referência, maior o valor da sua magnitude aparente. Valores de magnitude são medidos em várias bandas do espectro eletromagnético (ultra-violeta, infra-vermelho, vermelho, etc). Dessa forma, para um mesmo objeto, há várias medições de magnitude.

Uma particularidade encontrada nas bases de dados compiladas a partir de observações de magnitude coletadas por telescópios é a existência de medidas de erro associados a cada magnitude medida para um objeto espacial. Uma medida de erro indica a confiança na medição realizada para a magnitude correspondente. Os erros possuem relação com os dados de magnitude, tendo em vista que quanto menor a magnitude (i.e., quanto mais distante o objeto está), maior será o erro de medição correspondente [Fialho, 2020]. As Tabelas 1 e 2 ilustram essa particularidade.

Tabela 1: Valores de magnitudes nas bandas UGRIZ

ID	u	g	r	i	z
1237645942905110768	22.37	20.30	18.91	18.43	18.16
1237645942905569773	22.50	20.95	19.39	18.85	18.43
1237645943978328381	22.68	21.65	19.89	19.21	18.84
1237645943978524819	21.03	18.87	17.56	17.13	16.83
1237645943978524889	21.58	20.22	18.50	17.97	17.61

Tabela 2: Valores dos erros nas bandas UGRIZ

ID	err_u	err_g	err_r	err_i	err_z
1237645942905110768	0.31	0.05	0.05	0.06	0.08
1237645942905569773	0.54	0.12	0.08	0.08	0.11
1237645943978328381	0.53	0.15	0.09	0.09	0.11
1237645943978524819	0.18	0.04	0.05	0.06	0.07
1237645943978524889	0.26	0.07	0.06	0.06	0.08

Neste trabalho são exploradas técnicas de aprendizado supervisionado para o ajuste de modelos para estimar desvios fotométricos [D'Isanto and Polsterer, 2018]. Nosso interesse é investigar diferentes formas de incorporar no processo de aprendizado as medidas de erro associadas às magnitudes.

1.2 Motivação

Em Fialho [2020], foi proposta uma técnica para predição de desvio fotométrico usando modelos de redes neurais artificiais. O autor observou que as abordagens baseadas em Aprendizado de Máquina para predição de desvio fotométrico até então propostas desconsideravam os erros, ou tratavam medidas de erro e de magnitude como equivalentes. Nesse contexto, a abordagem de predição de desvio fotométrico proposta foi baseada no *Dropout*, uma técnica de regularização que não depende da modificação de uma função de custo, ou seja, modifica-se a própria rede neural [Srivastava et al., 2014]. O objetivo foi estimar desvios fotométricos considerando as medições de erro. A abordagem proposta usa as medições de erro para orientar o treinamento da rede neural, desativando o sinal de uma determinada banda de magnitude. Como resultado, a rede neural responsável por predizer os desvios fotométricos tem uma arquitetura adaptativa, as bandas de magnitudes tem uma probabilidade de serem descartadas dos exemplos com base na probabilidade de descarte. Essa probabilidade de descarte é calculada através da

predição dos erros das magnitudes utilizando as três estratégias desenvolvidas nesse trabalho.

Um dos passos da técnica proposta por Fialho [2020] envolve construir modelos de aprendizado de máquina para predizer o erro esperado de um determinado valor de magnitude. Contudo, uma limitação desse trabalho é que apenas uma estratégia de mapeamento de magnitudes para erros foi investigada. Em particular, a estratégia considerada mapeia um valor de magnitude para o erro esperado na banda correspondente. Neste presente trabalho, procuramos cobrir essa lacuna ao propor mais duas estratégias de mapeamento: (1) mapear um vetor de magnitudes para um vetor de erros esperados e (2) mapear um vetor de magnitudes para o erro esperado em uma banda específica.

1.3 Objetivos

Buscando dar continuidade à técnica de predição de desvio fotométrico apresentada em Fialho [2020], neste presente trabalho propomos a exploração de diferentes estratégias para estimar erros a partir de magnitudes, visando orientar de maneira mais precisa o processo de treinamento dos modelos de rede neural artificial desativando um sinal de magnitude em função de seu erro ou seus erros associados. Nosso objetivo geral é investigar modelos de aprendizado supervisionado para predição de desvios fotométricos. Como diferencial, são investigadas e comparadas três estratégias de mapeamento de magnitudes para erros. Dito isso, o trabalho possui os seguintes objetivos específicos:

1. Produzir modelos para predição de erros baseados nas magnitudes, utilizando três estratégias:
   (a) Estratégia muitos para muitos (i.e., usar um vetor de magnitudes para predizer um vetor de erros);
   (b) Estratégia muitos para um (i.e., usar um vetor de magnitudes para predizer um valor de erro);
   (c) Estratégia um para um (i.e., usar um valor de magnitude para predizer um erro esperado).
2. Utilizar os modelos produzidos para calcular a probabilidade de descarte de exemplos, com a finalidade de produzir modelos de predição de desvio fotométrico de maior poder preditivo.

1.4 Metodologia

Foram utilizadas as bases de leituras fotométricas publicamente disponibilizadas pelo Programa *Cosmostatistics Initiative* (COIN) e pelo portal do *SciServer* assim como apresentado no trabalho de Fialho [2020]. Os dados do COIN são retirados das amostras espectroscópicas coletadas do telescópio Sloan Digital Sky Survey (SDSS). Separados em duas subamostras: *COIN/HAPPY* (Happy), projetada com o intuito de isolar o efeito de cobertura espectroscópica limitada da amostra no espaço cor/magnitude, e *COIN/TEDDY* (Teddy), projetado para reproduzir o efeito de distintas distribuições de erros fotométricos e sua convolução com cobertura espacial cor/magnitude entre as amostras espectroscópicas e fotométricas. Os seguintes dados tornam possível encontrar, com diferentes técnicas de aprendizado de máquina, erros de magnitudes que influenciam na estimativa de redshifts, assim alcançando o objetivo de utilizar essa nova característica nos dados de entrada do estudo de estimativa de desvios fotométricos. As amostras em todos os bancos de dados apresentados possuem informações das magnitudes nas bandas (ugriz) e suas respectivas medidas de erros.

Para construção dos modelos, os dados foram separados em treino (80% do conjunto de dados) e teste (20% do conjunto de dados). Nos dados do conjunto de treino foi aplicada a técnica de validação cruzada de k camadas (cada camada com 20% do conjunto de treino). Os dados de Happy e Teddy possuem quatro subconjuntos separados. Para os experimentos apenas um foi utilizado para treino em quanto os outros para teste. No desenvolvimento dos modelos de aprendizado de máquina utilizamos a linguagem Python com as bibliotecas *Scikit-Learn*, *XGboost*, *Tensorflow* e outras. Na avaliação dos modelos é utilizada a métrica de erro médio quadrático *Mean Squared Error* (MSE).

1.5 Organização dos capítulos

O presente trabalho está organizado conforme descrição a seguir. O Capítulo 2 apresenta a fundamentação teórica. O Capítulo 3 apresenta a a descrição das propostas de mapeamento de magnitudes para erros esperados. O Capítulo 4 apresenta os experimentos computacionais realizados. O Capítulo 5 expõe as conclusões, com uma análise retrospectiva acerca dos resultados obtidos e das contribuições alcançadas, assim como apresenta descrições de possíveis continuações da pesquisa.

Capítulo 2

Fundamentação teórica

Esse capítulo tem como objetivo apresentar o conhecimento necessário para entender a fundamentação teórica e o contexto que engloba esse trabalho. Na Seção 2.1, são apresentados alguns conceitos astronômicos relacionados a esse trabalho. Já a Seção 2.2 apresenta a uma breve descrição dos métodos de regressão selecionados. Na Seção 2.3 é abordado o trabalho de Fialho [2020] o qual utilizamos como base para continuidade da pesquisa sobre estimativa de desvios fotométricos. Trabalhos que de alguma forma utilizam os erros durante o processo de ajuste de modelos para predição de desvio fotométrico são apresentados na Seção 2.4.

2.1 Conceitos da astronomia

2.1.1 Objetos Espaciais

Galáxias podem ser definidas como um amontoado de estrelas, gases e poeira ligados através de suas gravitações mútuas:

> *"A large assemblage of stars, gas, and dust bound together by their mutual gravitational attraction."* - [Kaufmann and Comins, 2014, pag. G-5]

O esquema de classificação desenvolvida por Edwin Hubble na década de 1920, possui as seguintes classificações: Espiral, Espiral Barrada, Elíptica, Lenticular, Irregular.

Galáxias espirais sem barra (normalmente) são caracterizadas por apresentarem um bojo central e caminhos arqueados de estrelas e poeira interestelar que parecem como "braços" que espiralam para fora do bojo central. Observações de várias galáxias espirais revelam quanto mais próximas as espirais maior o seu bojo central, a galáxia espiral mais próxima da terra é a galáxia de Andrômeda, M31 [Kaufmann and Comins, 2014, pag. 482].

A nossa galáxia, Via Láctea, é uma **galáxia espiral barrada**. Isto é, uma galáxia espiral com uma barra de estrelas e gases cruzando o bojo central. Essas barras são formadas em algumas galáxias de disco quando estrelas próximas do centro dessas galáxias que originalmente

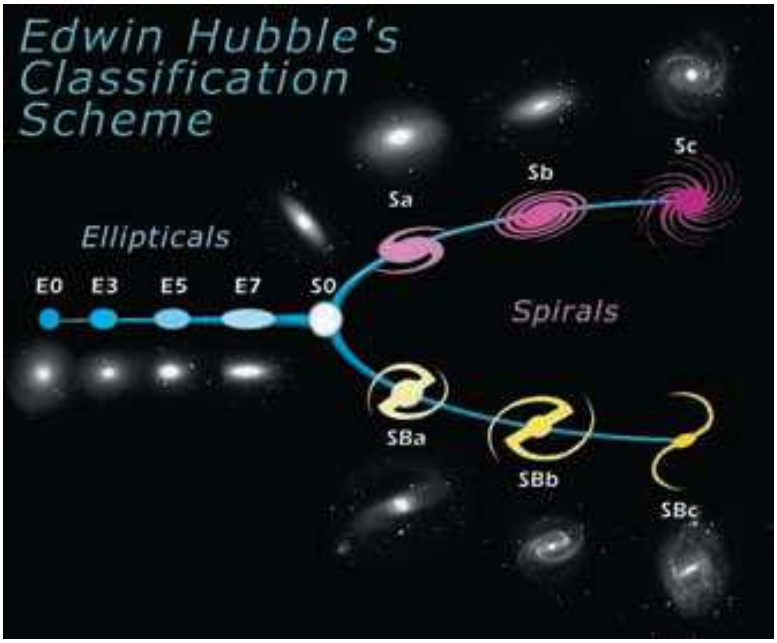

Figura 1: Esquema de classificação de galáxias elaborado por Hubble. Retirado de [Kaufmann and Comins, 2014, pag. 482].

realizavam uma órbita circular são forçadas a realizar uma órbita mais elíptica. Quando estrelas suficientes formam essas órbitas altamente elípticas, elas criam barras em galáxias espirais barradas. Essas estrelas depois atraem gases para a mesma órbita através da gravidade. Algum desses gases caem no buraco negro supermassivo que reside no centro de muitas (se não todas) galáxias espirais [Kaufmann and Comins, 2014, pag. 488].

Galáxias elípticas, nomeadas pela suas formas distintas, não possuem espirais. Ela apresentam uma enorme variedade de massa e tamanho, indo desdas menores e maiores galáxias do universo. Galáxias elípticas gigantes, contento em torno de 10 trilhões de massas solares, são mais raras, já as Galáxias elípticas anãs são muito mais comuns. Essas galáxias possuem poucas poeiras e gases interestelares, devido a isso sua população é constituída majoritariamente de estrelas mais velhas [Kaufmann and Comins, 2014, pag. 490].

Algumas dessa galáxias, barradas ou não barradas, não apresentam braços espirais e são semelhantes a lentes. Essas galáxias são chamadas de **galáxias lenticulares** [Kaufmann and Comins, 2014, pag. 488]. Galáxias lenticulares assim como as galáxias elípticas usaram ou perderam grande parte de sua poeira e gás interestelar resultando em uma baixa taxa de formação de estrelas [DeGraaff et al., 2007].

Edwin Hubble achou algumas galáxias que não podem ser classificadas como, espirais, espirais barradas, lenticulares ou elípticas. Elas geralmente são ricas em poeira e gases estelares e estrelas, velhas e novas. Exemplo de galáxias espirais são a Grande e Pequena Nuvem de

Magalhães, ambas podem ser vistas ao olho nu no hemisfério sul por serem bem próximas da Via Láctea [Kaufmann and Comins, 2014, pag. 490].

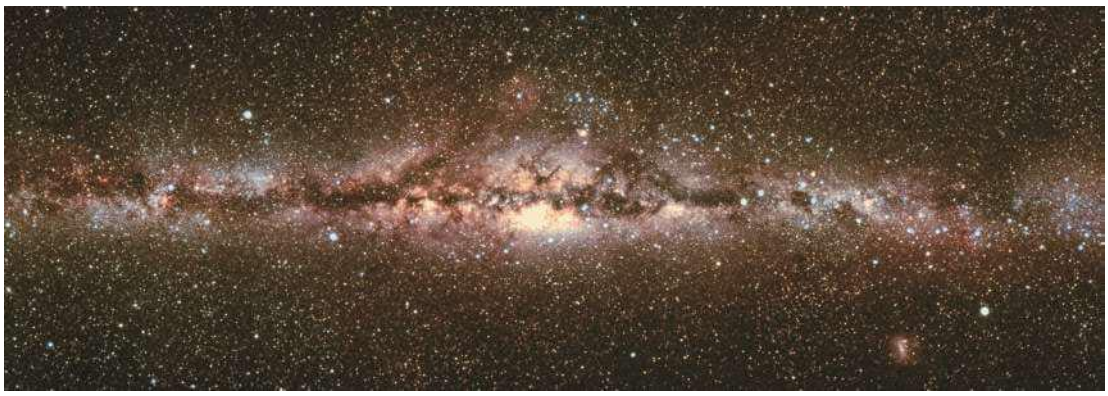

Figura 2: Imagem de metade da Via Láctea, galáxia em que os seres humanos habitam. Retirada de [Kaufmann and Comins, 2014, pag. 490].

Assim como as galáxias, existem vários tipos de **estrelas**. Grande parte das estrelas no céu são estrelas da sequência principal que por sua vez podem ser definidas como estrelas em equilíbrio hidrostático que fazem reações de fusão transformando hidrogênio em hélio em seus núcleos em taxas quase constantes [Kaufmann and Comins, 2014].

Quasares são objetos astronômicos parecidos com estrelas e emitem fortes ondas de rádio. Por conta disso, eles foram apelidados em inglês de *quasistellar radio sources*. Esse termo foi logo encurtado para Quasar. Quasares podem brilhar mais do que galáxias inteiras e apresentam grande *redshift*. Suas origens datam de 2 bilhões de anos após o Big Bang (grande explosão que originou o universo) e o número de detecções perto de 7 bilhões de anos trás já cai para próximo de 0 [Kaufmann and Comins, 2014, pag. 516].

Figura 3: Imagem do quasar, retirada de [Kaufmann and Comins, 2014, pag. 538].

2.1.2 Efeito Doppler

Como explicado no Capítulo 1, o efeito Doppler é observado quando uma onda tem seu comprimento alterado por causa do movimento de seu emissor em relação ao observador. Em caso de afastamento o comprimento de onda aumenta caracterizando um *redshift* no caso de aproximação o comprimento de onda diminui caracterizando um *blueshift*. Essa alteração do comprimento de onda se dá pela seguinte equação:

$$\frac{\Delta\lambda}{\lambda_0} = \frac{v}{c} \tag{2.1}$$

Onde λ_0 é o comprimento da onda gerado por um referencial estacionário, $\Delta\lambda$ é a diferença entre o comprimento de onda gerado e percebido dada por $\Delta\lambda = \lambda - \lambda_0$, λ é o comprimento de onda percebido, v é a velocidade do emissor em relação ao observador e c é a velocidade da luz no vácuo [Kaufmann and Comins, 2014, pag. 121].

2.1.3 Lei de Hubble

Em 1929, Edwin Hubble percebeu que o universo está em constante expansão através de observações de objetos distantes e cálculos de *redshift*. Através desses cálculos além de relatar a expansão do universo ele escreveu uma equação para essa expansão:

$$v = H_0 \cdot d \tag{2.2}$$

Onde v é a velocidade de afastamento, H_0 é a constante de Hubble e d é a distância até a Terra [Kaufmann and Comins, 2014, pag. 505].

2.1.4 Magnitudes

Existem distinções conceituais a respeito de magnitude, o brilho (ou magnitude) aparente de um objeto pode variar com a distância, definindo-se a *magnitude absoluta*, que é a magnitude a uma distância padrão de 10 parsecs (1 parsec = 3,26 anos-luz) [Fialho, 2020]. No trabalho é utilizada a magnitude aparente.

A escala de valores de magnitude é invertida, ou seja, quanto mais brilhante um objeto, menor será o seu valor de magnitude [Fialho, 2020] . Sendo assim, um corpo celeste de mag-

nitude aparente $m = +1mag$ possui seu brilho mais intenso que uma magnitude aparente de $m = +2mag$. Com a evolução da astronomia e redefinição de escalas[1], foram atribuídos valores negativos de magnitudes para objetos mais brilhantes. O Sol tem magnitude de $m = -26,7mag$. Já a Sírius, estrela mais brilhante do céu noturno, possui atualmente uma medida de magnitude aparente de -1,44.[2]

Existem vários sistemas de magnitude usados na Astronomia, com diferentes conjuntos de bandas. O mais famoso por muito tempo foi o sistema Johson (UBRVI). Com a relevância do levantamento SDSS, o sistema ugriz passou a dominar, sendo usado por vários outros levantamentos subsequentes como CFHTLS, DES, HSC, e LSST [Fialho, 2020].

Na Tabela 3 é apresentado as diferentes bandas de componentes do sistema ugriz, junto com os comprimentos médios de onda correspondentes a cada banda.[3]

Banda	$\overline{\lambda}$
u	365 nm
g	464 nm
r	658 nm
i	806 nm
z	900 nm

Tabela 3: Sistema de magnitudes ugriz. Para cada letra, é apresentado o comprimento de onda central, denotado por $\overline{\lambda}$.

Em ciência, uma medida sem seu respectivo erro, tem pouca utilidade. Outra informação disponível para cada objeto contido no catálogo SDSS é o erro associado a cada valor de magnitude medido. Como exemplo, a Tabela 4 apresenta os valores de magnitude e de erros para o objeto de identificador 587722981735792679. Como é de se esperar, quanto mais fraco o brilho do objeto em uma banda, maior o seu respectivo erro.

[1] com base no fluxo da estrela Vega

[2] Dados obtidos em `https://earthsky.org`.

[3] Informações do sistema ugriz https://astronomy.stackexchange.com.

Banda	Magnitude $\pm$ erro
Ultravioleta (u)	22,60 $\pm$ 0,73
Verde (g)	21,80 $\pm$ 0,20
Vermelho (r)	20,16 $\pm$ 0,05
Infravermelho - 7600 Å (i)	19,10 $\pm$ 0,04
Infravermelho - 9100 Å (z)	18,77 $\pm$ 0,08

Tabela 4: Para cada objeto no catálogo SDSS, são armazenados os valores de magnitude de acordo com o sistema ugriz, assim como os erros de medição correspondentes.

2.1.5 Desvio Fotométrico

Além da velocidade entre o emissor e o observador, o efeito Doppler pode ser causado pela própria expansão do espaço (universo). Nesse caso, a luz (uma onda eletromagnética) sofre um desvio para o vermelho, *redshift*. Para mensurar esse *redshift* podem ser utilizadas duas maneiras principais, a fotometria e a espectroscopia.

O desvio fotométrico ou *redshift* fotométrico é calculado através da medida do brilho de um objeto por meio de vários filtros. Filtros esses que são responsáveis por captar a intensidade de luz em uma determinada banda. Essa medida é importante para a Astronomia pois conseguimos medir a distância de objetos através da *Lei de Hubble*. O valor desse desvio também obedece a equação 2.1.

2.1.6 Levantamentos Fotométricos

A maneira típica de obter dados do universo é fazer levantamentos astronômicos que possuem o maior volume possível (área e profundidade) [Mickaelian, 2016]. Porém o problema da fotometria, especialmente de apenas 1 filtro, é que os objetos estão projetados de uma forma 2D no céu.

Para resolver esse problema podemos usar medidas espectrométricas dos objetos em questão, já que sabemos, pela lei de Hubble que o *redshift* de um objeto representa sua distância.

Todavia, nasce um novo problema. Levantamentos espectrométricos são muito mais demorados e custosos comparados aos fotométricos. Uma imagem (medida fotométrica) realiza em apenas alguns minutos o mapeamento de centenas de milhares de galáxias, uma medida espectroscópica realiza em horas o mapeamento de milhares de galáxias através de instrumentos multi-fibra.

Assim nasce a necessidade de uma maneira efetiva para obter dados precisos apenas com as medidas fotométricas. Essa maneira caracterizada pela medida fotométrica em várias bandas cobrindo uma ampla faixa de espectro. Apesar da medida não ser tão precisa quanto a de um espectrograma, ela atende as necessidades de diversas análises para esse tipo de levantamento.

Nesse trabalho, iremos utilizar os dados do SDSS (*Sloan Digital Sky Survey*), que é um levantamento astronômico fotométrico e espectrométrico. Para fotometria, ele utiliza Figuras multibanda profundas de um terço do céu. Essas bandas são denominada ugriz (U de 354nm, G de 476nm, R de 628nm, I de 769nm e Z de 925nm). Esses dados são liberados através de lançamentos anuais (*Data Release*) e atualmente ele se encontra no *Data Release* (DR) 17 [Abdurro'uf et al., 2022].

2.2 Métodos de regressão

Na área de estatística, a regressão é uma técnica capaz de inferir e quantificar relações entre variáveis dependentes e variáveis independentes. Essa seção visa apresentar uma visão geral dos métodos de regressão utilizados nesse trabalho. A Seção 2.2.1 apresenta as famosas redes neurais. A Seção 2.2.2 apresenta as arvores de decisão, muito utilizadas para problemas de regressão. Na Seção 2.2.3 mostra a base teórica do algoritmo k-nearest neighbors. Na Seção 2.2.4 é apresentado o algoritmo XGBoost utilizado por competições de machine learning como método base.

2.2.1 Redes Neurais MLP

As MLP são consideradas uma derivação de RNAs (Redes Neurais Artificiais), uma vez que envolvem mais de uma camada oculta no processo de modelagem, sendo considerada uma estratégia em aprendizado de máquina. Uma das vantagens de usar técnicas de aprendizado de máquina é a capacidade de capturar características, mesmo quando as distribuições de probabilidade são desconhecidas [Inocente et al., 2022].

Uma Rede Neural Artificial pode ser definida como uma combinação massivamente paralela de uma unidade de processamento simples que pode adquirir conhecimento do ambiente por meio de um processo de aprendizado e armazenamento do conhecimento em suas conexões [Guresen and Kayakutlu, 2011].

A rede neural perceptron de multicamada (MLP) é uma das mais implementadas. Em ter-

mos de habilidades de mapeamento, acredita-se que a MLP possui a capacidade de aproximar funções arbitrárias - [Guresen et al., 2011] e [Mukhopadhyay, 2003]. Isso tem sido importante no estudo da dinâmica não linear e outros problemas de mapeamento de funções. Duas características importantes do perceptron multicamadas são:

1. Elementos de processamento não lineares (PEs) que possuem uma não linearidade que deve ser suave (a função logística e a tangente hiperbólica são as mais utilizadas);

2. Interconectividade, ou seja, qualquer elemento de uma determinada camada pode alimentar todos os elementos das possíveis camadas seguintes

Através desse algoritmo é possível aplicar regressão linear para aproximar os valores de entrada dos valores de saída da Rede.

2.2.2 Árvore de Decisão

As árvores de decisão são bem conhecidas como meios de representação do conhecimento e como algoritmos para resolver vários problemas de otimização combinatória e geometria computacional [Azad et al., 2021]. Os algoritmos são um método de aprendizado supervisionado não paramétrico muito utilizado para problemas de classificação e regressão [Fletcher and Islam, 2019]. Eles são treinados em dados rotulados para classificar corretamente dados não vistos anteriormente e não fazem suposições sobre a distribuição de dados subjacentes. Possuem vantagens sobre outros tipos de métodos de aprendizado supervisionado o que os tornam atraentes para os cientistas de dados.

Um exemplo é o algoritmo de árvore de decisão impulsionada chamado ArborZ, utilizado para estimar redshifts de galáxias inacessíveis [Gerdes et al., 2010], e o algoritmo de árvore de regressão imparcial [Hothorn et al., 2006] utilizado em procedimentos de inferência condicional aplicável a todos os tipos de problemas de regressão.

2.2.3 K Nearest Neighbor

O Algoritmo K Nearest Neighbor foi desenvolvido a partir da necessidade de realizar análises discriminantes quando estimativas paramétricas confiáveis de densidades de probabilidade são desconhecidas ou difíceis de determinar [Peterson, 2009]. Em 1967, algumas das propriedades formais da regra do k-vizinho mais próximo foram elaboradas; exemplo disso foi

demonstração de que para $k = 1$ e $n \implies \infty$ o erro de predição do k-vizinho mais próximo é limitado acima por duas vezes a taxa de erro de Bayes [Cover and Hart, 1967]. Uma vez que tais propriedades formais da classificação de k-vizinhos mais próximos foram estabelecidas, uma longa linha de investigação se seguiu incluindo novas abordagens de rejeição [Hellman, 1970] e refinamentos em relação à taxa de erro de Bayes [Fukunaga and Hostetler, 1975], além de outros citados por Peterson.

2.2.4 XGBoost

O sistema XGBoost é um sistema de árvores de decisão (ver Seção 2.2.2) escalável e flexível utilizado em competições de ciência de dados para conseguir resultados no estado da arte. O XGBoost além de conseguir excelente resultados, ele requer muito menos recursos por causa de suas funcionalidades [Chen and Guestrin, 2016]:

1. Padrões de Acesso ao Cache
2. Computação Paralela
3. Compressão de Dados
4. Fragmentação de Dados

2.2.5 Random Forest

A unidade básica de Random Forest é uma árvore binária construída utilizando particionamento recursivo. Um método cuja as divisões binárias dividem recursivamente a árvore em nós terminais homogêneos ou quase homogêneos (as extremidades da árvore). Uma boa divisão binária envia dados de um nó de árvore pai para seus dois nós filhos para que a homogeneidade resultante nos nós filhos seja melhor que a encontrada no nó pai [Chen and Ishwaran, 2012].

2.2.6 Isotonic Regression

A regressão isotônica é uma técnica de ajuste linear que se utiliza de uma sequência de observações com caráter ascendente fazendo a curva de um modelo passar o mais próximo possível das observações [Barlow, 1972].

2.3 Utilizando erros de magnitudes para predizer desvios fotométricos

Fialho [2020] investigou a predição de desvios fotométricos por meio de modelos de rede neural com *Dropout*. A técnica de *Dropout* consiste em alterar aleatoriamente a arquitetura de uma rede neural de forma a minimizar os riscos de *overfitting*. Essa alteração é feita com uma probabilidade calculada através do erro observado e do erro esperado das bandas. O cálculo da diferença entre o erro observado, do erro esperado e da probabilidade de *dropout* são definidos pelas Equações 2.3 e 2.4 respectivamente. A primeira representa a diferença normalizada entre **erro observado** e **erro esperado**. A segunda representa probabilidade de descartar uma magnitude.

$$\delta_{ij} = \frac{\varepsilon_j^{(i)} - h_j(m_j^{(i)})}{\varepsilon_j^{(i)}} \tag{2.3}$$

Onde δ_{ij} é a diferença normalizada entre o erro observado e esperado da banda j da amostra i, ε_{ij} é o erro observado da banda j da amostra i, h_j é função do regressor da banda j e m_{ij} é o valor magnitude da banda j da amostra i.

$$p_{ij} = 1 - \frac{e^{\delta_{ij}}}{\sum_{k=1}^{b} e^{\delta_{ik}}} \tag{2.4}$$

Onde p_{ij} é a probabilidade de descarte da magnitude da banda j da amostra i, ε_{ij} é definido na Equação 2.3, b é a quantidade de bandas, k é o índice de uma banda e ε_{ik} é o erro observado da banda k da amostra i.

Para produzir $\tilde{\mathbf{m}} = [\tilde{m}_1, \tilde{m}_2, \ldots, \tilde{m}_b]$, a versão mascarada de um exemplo de treinamento $\mathbf{m} = [m_1, m_2, \ldots, m_b]$ associado a um vetor de erro $\varepsilon = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_b]$ [Fialho, 2020], suponha que $r_{ij} \sim \text{Bernoulli}(p_{ij})$ indica um valor amostrado de uma distribuição Bernoulli com parâmetro p_{ij}, ou seja, $r_{ij} \in \{0,1\}$. Para calcular $\tilde{m}_j^{(i)}$ $1 \leq j \leq b$, usamos a Equação 2.5.

$$\tilde{m}_j^{(i)} = m_j^{(i)} \times r_{ij} \tag{2.5}$$

Considerando que as máscaras computadas para $\mathbf{m}^{(i)}$ estão organizadas em um vetor denotado por $\mathbf{r}^{(i)}$, pode-se reescrever o processo de alteração aplicado a $\mathbf{m}^{(i)}$ conforme a Equação 2.6. Nessa Equação, $\odot$ representa o produto de Hadamard[4].

[4]O produto de Hadamard é uma operação binária que toma dois vetores de mesmas dimensões como operandos e produz outro vetor de mesma dimensão dos operandos, onde cada elemento i é o produto dos elementos i dos

$$\tilde{\mathbf{m}}^{(i)} = \mathbf{m}^{(i)} \odot \mathbf{r}^{(i)} \tag{2.6}$$

2.3.1 Exemplo numérico

Nesta seção é fornecido um exemplo numérico utilizado por Fialho [2020] para sua proposta de aplicação. A Figura 4 apresenta uma visão geral do exemplo para o cálculo de *dropout* e como ele é aplicado aos exemplos de entrada, seguindo um fluxo ordenado de transformações. Na figura retirada de Fialho [2020], os círculos numerados identificam a sequência de passos para transformação de um exemplo de treinamento em particular.

AQUI

dois vetores originais. Essa operação é também definida para matrizes em geral.

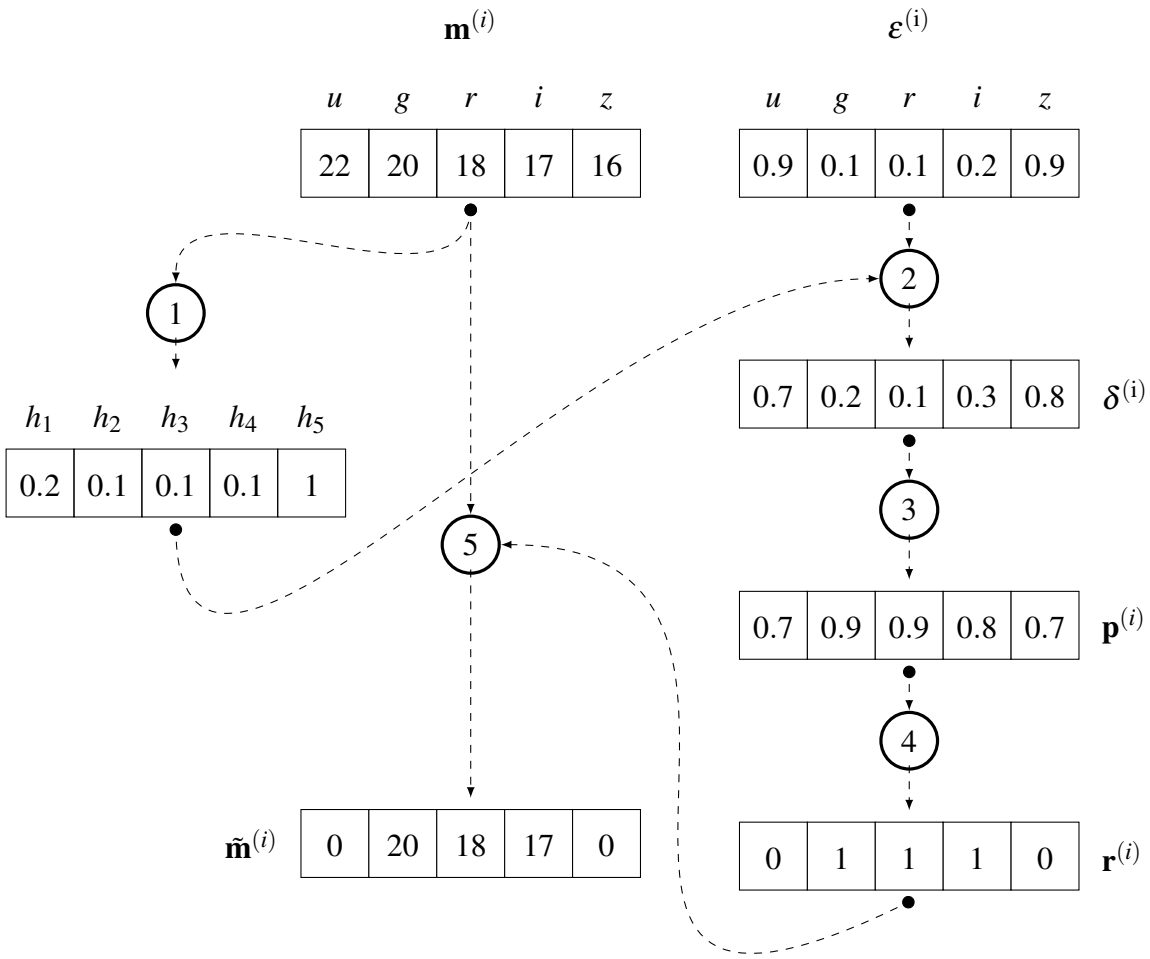

Figura 4: O passo 1 computa os erros esperados para cada valor de magnitude. O passo 2 aplica a Equação 2.3 para produzir o vetor de desvios $\delta^{(i)}$. O vetor de probabilidades $\mathbf{p}^{(i)}$ é computado no passo 3 por meio da Equação 2.4. No passo 4, o vetor aleatório de máscaras ($\mathbf{r}^{(i)}$) é computado. O passo 5 computa o vetor $\tilde{\mathbf{m}}^{(i)}$) aplicando $\mathbf{r}^{(i)}$ ao vetor de magnitudes original, o que resulta no exemplo alterado a ser usado no treinamento. Figura retirada de Fialho [2020].

2.4 Trabalhos relacionados

Para selecionar os trabalhos relacionados foi realizada uma busca no Scopus com a seguinte query "photometric AND redshift AND estimation AND deep OR machine AND learning". Nela são selecionados todos os artigos que contém as palavras "photometric", "redshift", "estimation" e "deep learning" ou "machine learning". Para manter um bom nível de qualidade da literatura selecionada, escolhemos os artigos por ordem de maior número de citação e mais recentes, priorizado o número de citações.

O trabalho D'Isanto and Polsterer [2018] realiza estimações de *redshift* através da criação sintética de características utilizando combinações de magnitudes, erros, raios e elipticidades de quasares, retirados dos catálogos SDSS-DR7 e SDSS-DR9, a fim de melhorar a taxa de predição dos modelos de progressão e classificação. Sendo assim esse trabalho utiliza as informações dos erros das magnitudes para criar essas características sintéticas.

Já Hoyle [2016] utiliza modelos baseados de árvores de decisão (AdaBoost) e redes neurais profundas para estimar os desvios fotométricos com base no catálogo de galáxias disponibilizado pelo SDSS-DR10. As características utilizadas no trabalho foram os valores das magnitudes e das cores. Os experimentos desse trabalho utilizam apenas as informações das magnitudes e das cores, assim todos os dados relacionado aos erros foram excluídos.

Capítulo 3

Estratégias de regressão de erros por magnitudes

Esse capítulo tem como objetivo apresentar as estratégias utilizadas nesse trabalho para predição dos erros a partir de magnitudes. Na Seção 3.1, são apresentadas as abordagens existentes para uso das medições de erro durante a predição de desvio fotométrico. A Seção 3.2 apresenta as estratégias de mapeamento de magnitudes para erros. A Seção 3.3 apresenta as adaptações necessárias na equação que computa os desvios dos erros.

3.1 Abordagens para uso dos erros das medições

Existem várias alternativas para construir modelos de aprendizado de máquina para predição de desvio fotométrico, no que diz respeito aos erros medidos para cada banda de magnitude. Uma delas utilizar apenas os dados de magnitude como variáveis preditoras. Uma desvantagem dessa abordagem é descartar os erros que podem conter informações úteis para predição.

Outra abordagem consiste em utilizar os erros juntos das informações de magnitudes e considerar todas essas medições como variáveis preditoras. Uma desvantagem desta abordagem é que o algoritmo de aprendizado fica responsável por identificar eventuais relações de dependências entre esses dois grupos de variáveis preditoras, o de magnitudes e o de erros.

Uma terceira abordagem é ainda considerar as medições de erros, mas modelar de forma explícita as relações de dependência eventualmente existentes entre magnitudes e erros.

Neste trabalho, adotamos a terceira abordagem. Em particular, investigamos a construção de modelos que consigam predizer erros a partir de magnitudes para que essa relação possa ser posteriormente utilizada na predição do desvio fotométrico. Investigamos três estratégias para produção desses modelos de mapeamento, que são descritas na próxima seção.

3.2 Estratégias de mapeamento de magnitudes para erros

Nesse trabalho, são investigadas três estratégias para realizar o mapeamento de magnitudes para erros. Uma dessas estratégias foi proposta por Fialho [2020], e utiliza o valor da magnitude

de uma banda para predizer o erro correspondente, 1×1. As duas outras estratégias são propostas neste trabalho e consistem em utilizar as informações de todas as bandas para predizer todos os erros $m \times m$ e utilizar as informações das bandas para predizer apenas um erro $m \times 1$.

Considere um conjunto de dados $D = \{(\mathbf{m}^{(i)}, \varepsilon^{(i)})\}, 1 \leq i \leq |D|\}$. Nesse conjunto, $\mathbf{m}^{(i)}$ é um vetor cujos componentes são medições de magnitudes para o i-ésimo objeto, conforme a Equação 3.1.

$$\mathbf{m}^{(i)} = \left[m_1^{(i)}, m_2^{(i)}, \ldots, m_b^{(i)}\right] \tag{3.1}$$

Já $\varepsilon^{(i)}$ é outro vetor cujos componentes são medições de erros nas diferentes bandas de magnitudes, conforme a Equação 3.2.

$$\varepsilon^{(i)} = \left[\varepsilon_1^{(i)}, \varepsilon_2^{(i)}, \ldots, \varepsilon_b^{(i)}\right] \tag{3.2}$$

Nas equações acima, $m_j^{(i)}$ representa a j-ésima magnitude medida para o i-ésimo objeto, $\varepsilon_j^{(i)}$ corresponde ao valor do erro correspondente e b o número de bandas.

Considere que $\mathbf{m}$ e ε são vetores quaisquer de magnitudes e de erros, respectivamente. As estratégias de mapeamento de magnitudes para erros aqui apresentadas envolvem usar um subconjunto das componentes de $\mathbf{m}$ como variáveis preditoras e um subconjunto das componentes de ε como variáveis alvo.

3.2.1 Estratégia um para um (1×1)

Essa estratégia envolve construir um modelo que utiliza uma magnitude como variável preditora e apenas um erro como variável dependente. Nessa estratégia, é gerado um modelo de mapeamento para cada banda de magnitude. Consequentemente, b modelos de mapeamento são produzidos.

A estratégia de mapeamento um-para-um é ilustrada na Figura 5. A partir de uma banda de magnitude, uma única estimativa de erro é gerada. Essa foi a estratégia adotada por Fialho [2020]. Em nossos experimentos, comparamos as três estratégias.

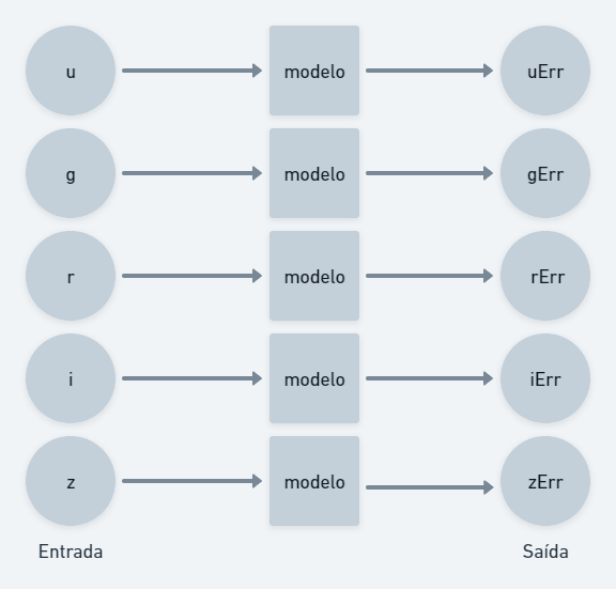

Figura 5: Ilustração da estratégia um para um.

3.2.2 Estratégia muitos para muitos ($m \times m$)

Essa estratégia envolve construir um modelo de regressão múltipla que utiliza todas as magnitudes como variáveis preditoras e todos os erros como variáveis dependentes. Essa estratégia gera apenas um modelo de mapeamento. Mais especificamente, o modelo de regressão múltipla gerado utiliza como entrada um vetor de magnitudes e gera como resultado um vetor de erros esperados (previstos).

A Figura 6 ilustra esquematicamente a estratégia muito-para-muitos. A estratégia adota como entrada que representa as cinco bandas de magnitudes presentes com o objetivo de estimar e retornar como saída a estimativa simultânea de cada erro de banda.

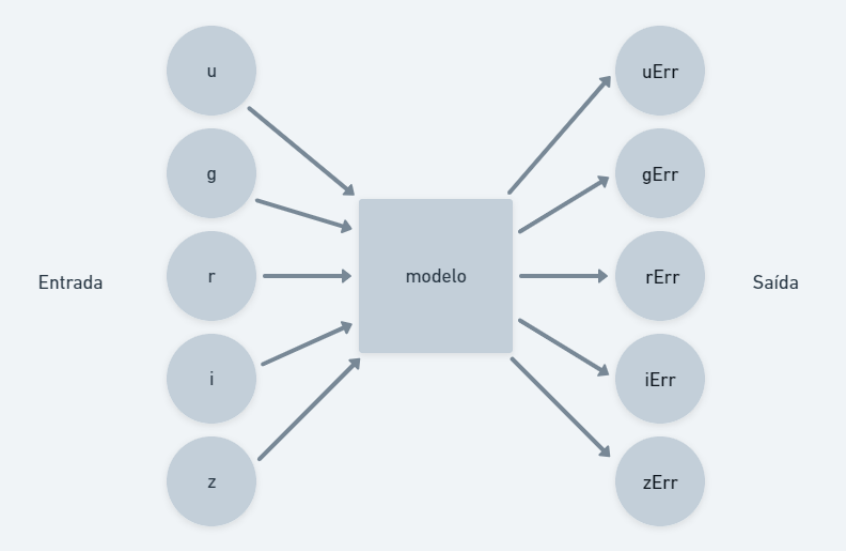

Figura 6: Ilustração da estratégia muitos para muitos.

3.2.3 Estratégia muitos para um ($m \times 1$)

Essa estratégia corresponde a construir um modelo que utiliza todas as magnitudes como variáveis preditoras e o erro em uma banda específica como variável dependente. Nessa estratégia, é gerado um modelo de mapeamento para cada banda de magnitude. Consequentemente, b modelos de mapeamento são produzidos.

A estratégia $m \times 1$ é ilustrada na Figura 7. Repare que todos os valores de magnitudes (Entrada) são usados como preditores para um único alvo (Saída), que é o erro esperado em uma banda específica.

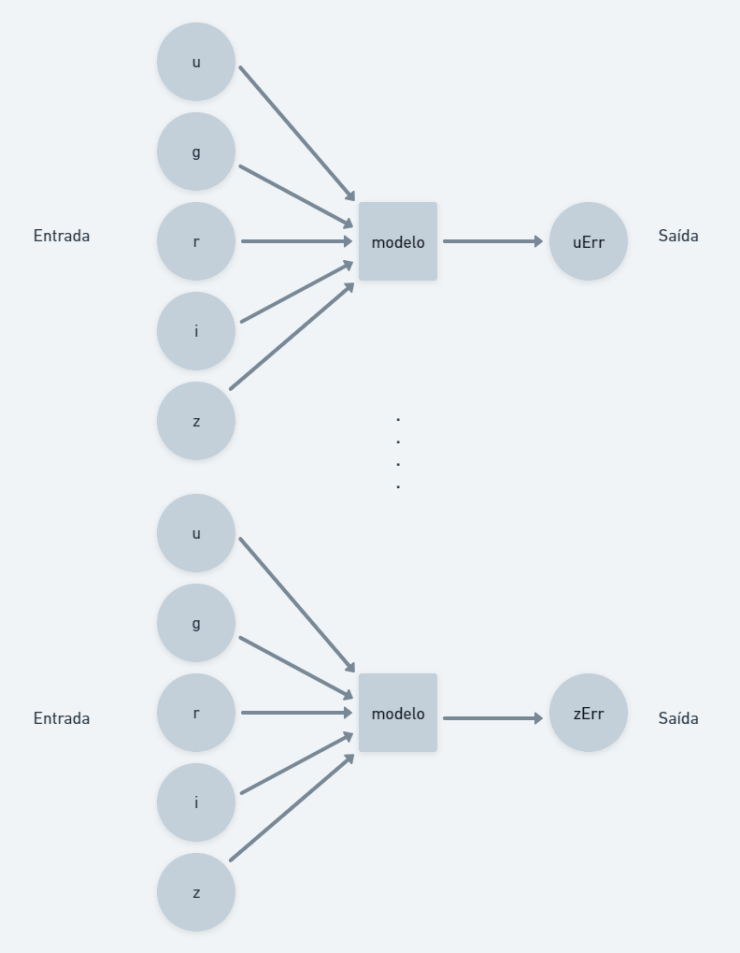

Figura 7: Ilustração da estratégia muitos para um.

3.3 Predição de desvio fotométrico

O presente trabalho estende o apresentado em Fialho [2020] (Seção 2.3) na medida em que investiga novas estratégias de regressão para computar a estimativa h_j e consequentemente δ_{ij}. Tendo em vista as novas estratégias, é necessário fazer duas versões da Equação 2.3 para atender as estratégias $m \times m$ e $m \times 1$. A Equação 3.3 é uma adaptação da Equação 2.3 para a estratégia $m \times m$.

$$\delta_i = \frac{\varepsilon^{(i)} - h(\mathbf{m}^{(i)})}{\varepsilon^{(i)}} \tag{3.3}$$

Na equação acima, δ_i é o vetor de diferença de erros da amostra contendo bs δ_{ij}, $\varepsilon^{(i)}$ é o vetor dos erros observados e h é função do regressor que recebe como entrada um vetor de magnitudes e retorna um vetor de erros esperados para a amostra i. A Equação 3.4 é uma adaptação da Equação 2.3 para estratégia muitos-para-1.

$$\delta_{ij} = \frac{\varepsilon_j^{(i)} - h_j(\mathbf{m}^{(i)})}{\varepsilon_j^{(i)}} \tag{3.4}$$

Na equação acima, $\varepsilon_j^{(i)}$ é o erro na banda j na amostra i, h_j é a função do regressor que recebe um vetor de magnitudes e retorna um escalar correspondente ao erro esperado na banda j e $\mathbf{m}^{(i)}$ é o vetor de magnitudes medidas para o i-ésimo objeto.

Capítulo 4

Experimentos

Esse Capítulo tem como objetivo documentar o ambiente de experimentos, os conjunto de dados e análises realizados no mesmo, assim como os resultados dos modelos de regressão. Na Seção 4.1 são apresentados as condições de software e hardware que os experimentos foram realizados para sua reprodutibilidade. Na Seção 4.2 são apresentadas os conjuntos de dados utilizados no experimento, assim como suas informações relevantes. Na Seção 4.3 é relatada as etapas para tratamento e geração dos modelos. Finalmente, na Seção 4.4 é apresentado o resultado dos modelos de regressão para os conjuntos de dados Teddy e Happy.

Os algoritmos de aprendizado de máquina processam os dados provenientes do Teddye Happy, percorrendo cada banda de magnitude (u,g,r,i,z) gerando um erro para ser utilizado nas métricas de estimação de desvios fotométricos.

4.1 Configurações de hardware e software

A implementação da solução desse trabalho utiliza a linguagem Python [Van Rossum and Drake, 2009] na versão 3.10.5. Os principais pacotes utilizados foram o Scikit-learn [Pedregosa et al., 2011] e o XGBoost [Chen and Guestrin, 2016], como pode ser consultado no repositório do projeto nos arquivos *.tools-version* (versão do Python) e *requirements.txt* (versão dos pacotes utilizados). Para executar os experimentos, utilizamos uma máquina com as especificações de hardware e software apresentadas na Tabela 5.

Memória RAM	16GB
CPU	AMD Ryzen 5 3600 6-Core Processor
GPU	TU116 [GeForce GTX 1660]
Linux Distro	openSUSE Tumbleweed 20220729
Linux Kernel	5.18.11-1-default

Tabela 5: Especificações de hardware e software da bancada de experimentos

4.2 Conjuntos de dados

Para realizar os experimentos, foram selecionados dados de levantamentos fotométricos nas bandas ugriz. Mais especificamente os bancos de dados compartilhado pelo COIN (Teddy e Happy) [Beck et al., 2017]. A Tabela 6, mostra o formato dos dados de interesse.

Campo	Tipo	Decrição
u	float	banda ultravioleta
g	float	banda verde
r	float	banda vermelha
i	float	banda infravermelha
z	float	banda z-infravermelha
err_ u	float	erro da banda u
err_ g	float	erro da banda g
err_ r	float	erro da banda r
err_ i	float	erro da banda i
err_ z	float	erro da banda z
redshift	float	redshift observado
err_ redshift	float	erro no redshift observado

Tabela 6: Campos de interesse dos conjunto de dados.

4.2.1 Teddy

O conjunto de dados Teddy é dividido em quatro partes, três são separadas apenas para teste e a primeira que será trabalhada o treinamento. O número de variáveis nesse conjunto de dados é de 13 (contando com o ID do objeto), nesse conjunto de dados há um total de 74309 observações como apresentado nas tabelas 7 e 8 junto com algumas propriedades estatísticas.

Tabela 7: Propiedades Estatísicas do conjunto de dados (teddy_data)

	u	g	r	i	z
count	74309.000	74309.000	74309.000	74309.000	74309.000
mean	21.883	20.154	18.590	17.989	17.622
std	0.896	0.973	0.856	0.786	0.774
min	17.712	16.491	15.178	14.744	14.416
25%	21.200	19.387	17.934	17.390	17.039
50%	21.925	20.253	18.611	18.014	17.645
75%	22.538	20.904	19.177	18.532	18.148
max	24.874	22.900	20.997	20.863	20.547

Tabela 8: Propiedades Estatísicas do conjunto de dados (teddy_data)

	err_u	err_g	err_r	err_i	err_z	redshift	err_redshift
count	74309.000	74309.000	74309.000	74309.000	74309.000	74309.000	74309.000
mean	0.371	0.076	0.066	0.076	0.107	0.323	0.001
std	0.228	0.035	0.018	0.016	0.027	0.088	0.049
min	0.018	0.024	0.035	0.043	0.049	0.011	0.000
25%	0.198	0.048	0.054	0.065	0.089	0.251	0.000
50%	0.320	0.068	0.062	0.072	0.102	0.322	0.000
75%	0.485	0.094	0.073	0.083	0.120	0.385	0.000
max	2.534	0.435	0.293	0.334	0.514	0.940	7.789

Analisando as relações através de gráficos, pode-se observar que no geral há uma correlação entre os erros e as magnitudes. O gráfico na figura 8 mostra as relações das bandas entre si. Nele podemos ver como elas tem uma alta relação, e além disso na diagonal temos o gráfico de distribuição da amostra.

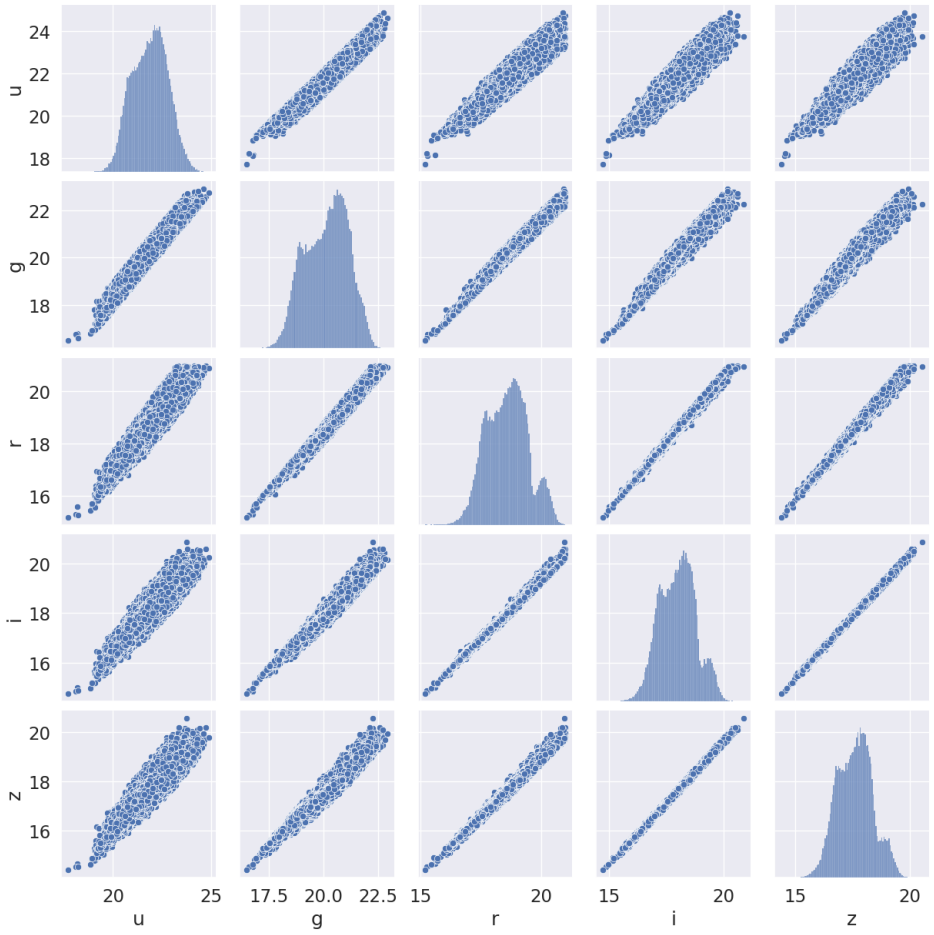

Figura 8: Bandas X Bandas (Teddy)

Seguindo a mesma lógica da figura acima, a figura 9 ilustra as mesma propriedades só que para os valores dos erros das bandas. Apesar da relação ser menor, podemos notar que ela ainda está presente, também é apresentado a distribuição na diagonal da figura.

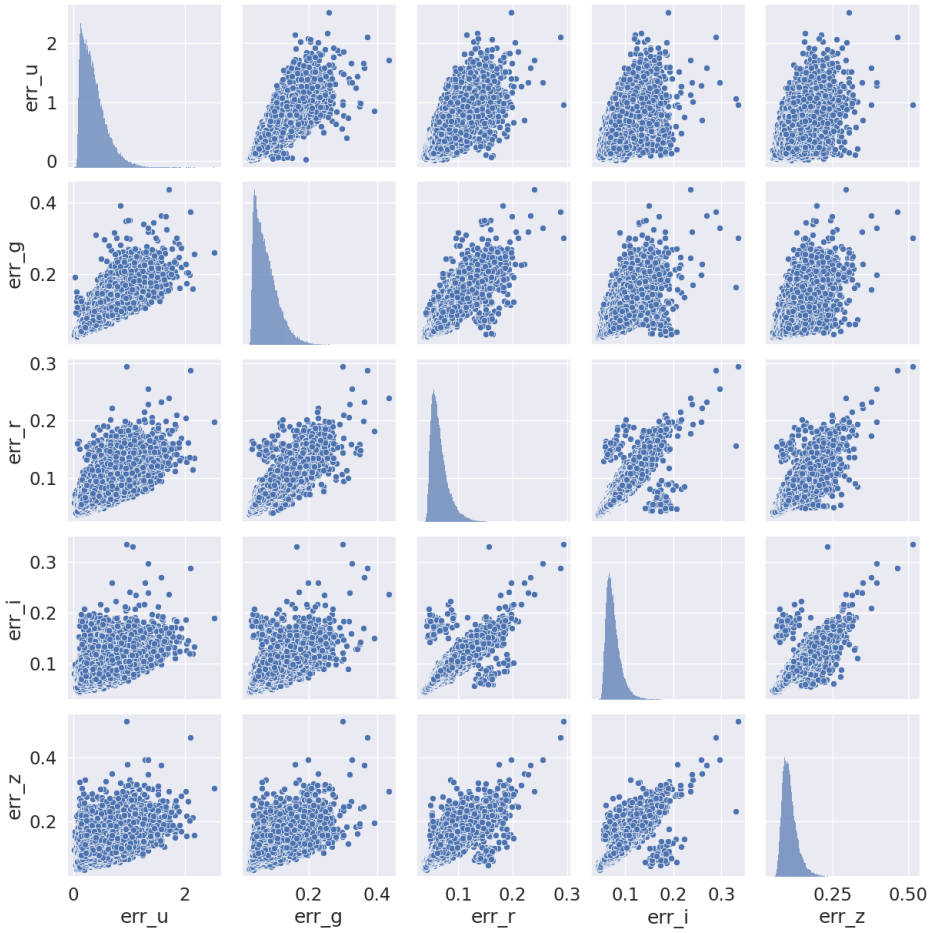

Figura 9: Erros X Errors (Teddy)

Finalizando a figura 10 exibe a relação entre as bandas e os erros, nesse caso a diagonal principal não mostra mais a distribuição, mas a relações entre as duas variáveis. Apesar de algumas relações entre as bandas e seus respectivos erros ser explicitas ao olhar nu, a seguir temos um gráfico de regressão linear para explorar e documentar cada uma delas.

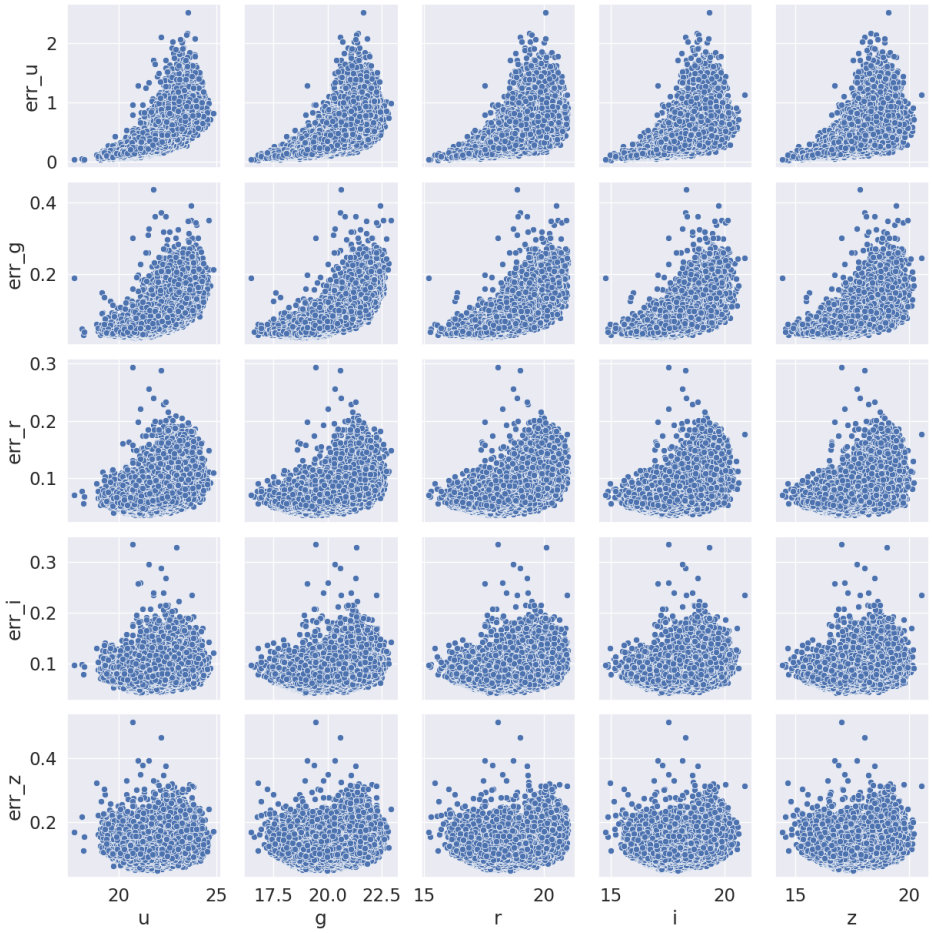

Figura 10: Bandas X Errors (Teddy)

A figura 11 exibe o a relação entre os valores da banda U e seus erros. Nela é notável a relação entre eles e a linha de regressão apenas corrobora com esse fato.

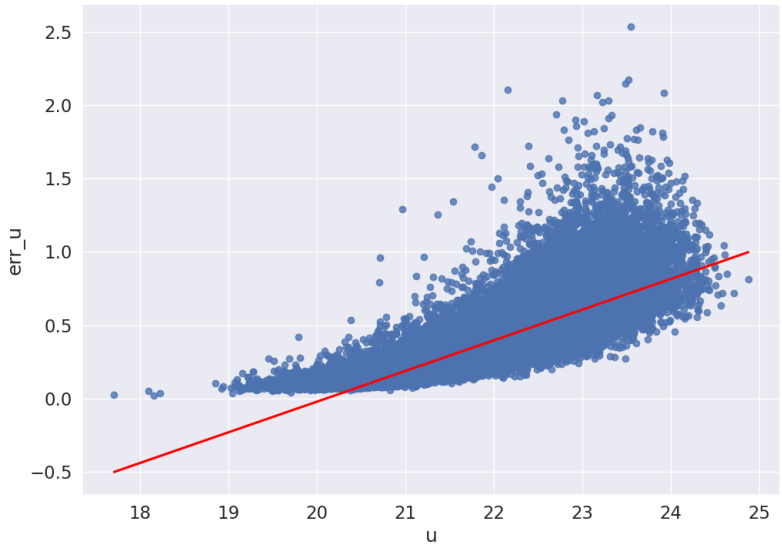

Figura 11: Banda U: Magnitude X Erro (Teddy)

Da mesma forma, a figura 12 ilustra a relação da banda G com seu erro junto com uma linha de regressão. Sendo um caso similar, a relação é notável ao olhar e a linha apenas retifica com o fato.

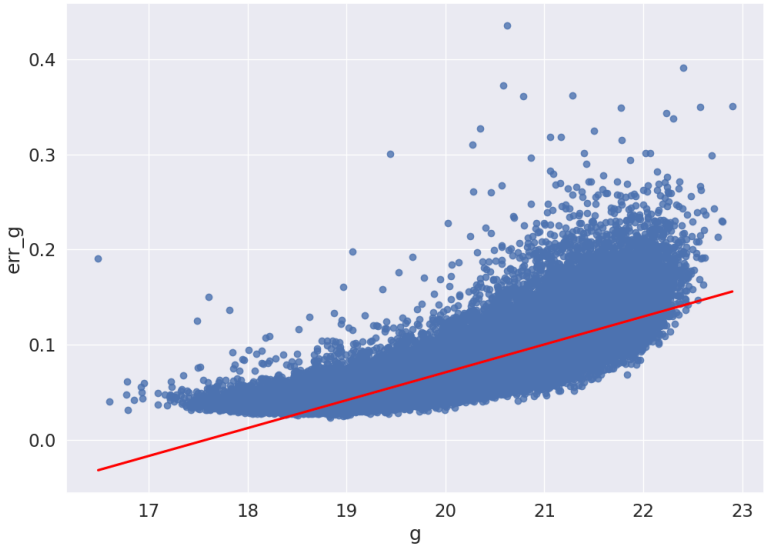

Figura 12: Banda G: Magnitude X Erro (Teddy)

Seguindo o raciocínio, a figura 13 apresenta a relação entre a banda R e seu erro. Assim como nos outros casos, quanto maior o valor da banda maior o erro.

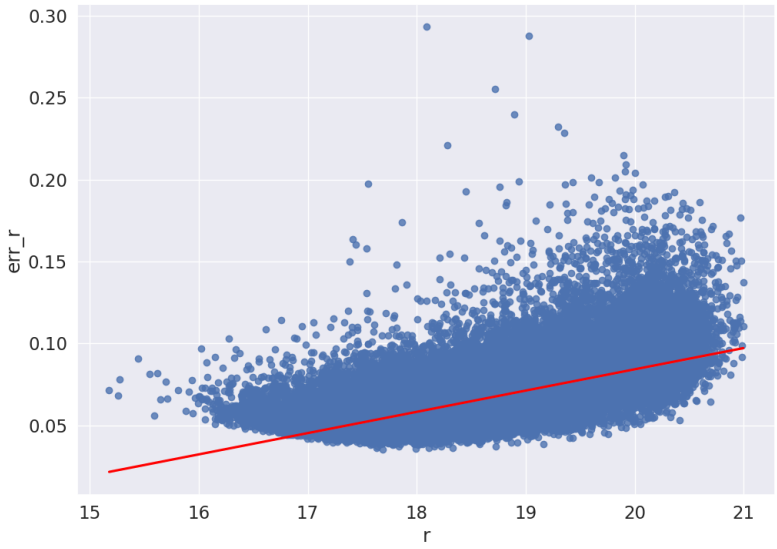

Figura 13: Banda R: Magnitude X Erro (Teddy)

A seguir, a figura 14 expõe a relação entre os valores da banda I e seu erro. Nesse caso, não fica tão óbvio ao olhar a relação, porém a linha de regressão revela esse fato.

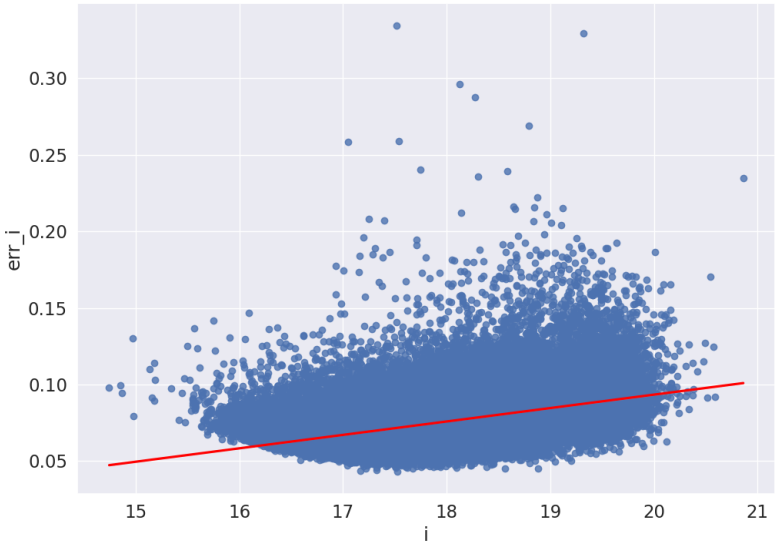

Figura 14: Banda I: Magnitude X Erro (Teddy)

Finalizando, a figura 15 evidencia a relação entre a banda Z e seu erro. Assim como no caso da figura anterior, não fica óbvio ao olhar a relação entre eles, porém a curva de regressão mais uma vez confirma esse fato.

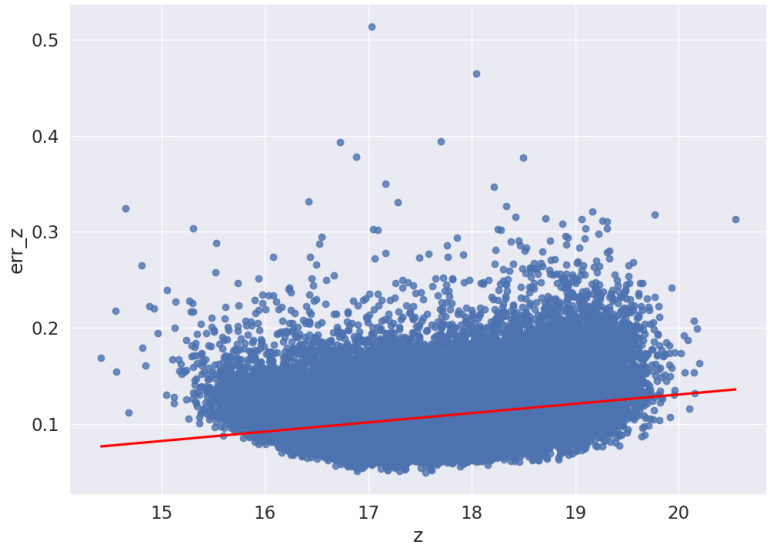

Figura 15: Banda Z: Magnitude X Erro (Teddy)

4.2.2 Happy

Assim como o Teddy, o Happy também é dividido em quatro partes, três para teste e uma para o treinamento dos modelos. Como as propriedades desse conjuntos são semelhantes, ele também apresenta 13 variáveis (contando com ID do objeto). Nele são presentes 74950 observações como exposto junto com algumas propriedades estatísticas nas tabelas 9 e 10.

Tabela 9: Propiedades Estatísicas do conjunto de dados (happy_data)

	u	g	r	i	z
count	74950.000	74950.000	74950.000	74950.000	74950.000
mean	22.089	20.236	18.911	18.224	17.865
std	2.359	2.023	1.680	1.450	1.407
min	11.670	10.544	10.713	10.602	10.678
25%	20.025	18.322	17.451	17.039	16.749
50%	22.350	20.871	19.229	18.552	18.161
75%	23.846	21.934	20.386	19.457	19.019
max	31.475	28.652	27.929	28.221	29.105

Tabela 10: Propiedades Estatísicas do conjunto de dados (happy_data)

	err_u	err_g	err_r	err_i	err_z	redshift	err_redshift
count	74950.000	74950.000	74950.000	74950.000	74950.000	74950.000	74950.000
mean	0.489	0.129	0.093	0.094	0.141	0.351	0.000
std	0.457	0.196	0.101	0.098	0.122	0.210	0.014
min	0.012	0.022	0.034	0.041	0.048	0.000	0.000
25%	0.081	0.039	0.055	0.068	0.099	0.135	0.000
50%	0.383	0.093	0.078	0.085	0.124	0.374	0.000
75%	0.765	0.175	0.118	0.107	0.158	0.529	0.000
max	11.780	32.901	8.897	13.248	9.394	1.472	2.779

Com uma análise de relações feita a partir de imagens de gráficos, observa-se que há uma tendência na relação entre os valores de erros e magnitudes também nesse conjunto de dados. O gráfico na figura 8 expõe as relações banda a banda. Dele pode-se concluir que entre si as bandas tem uma alta relação e na diagonal é observado a distribuição da amostra.

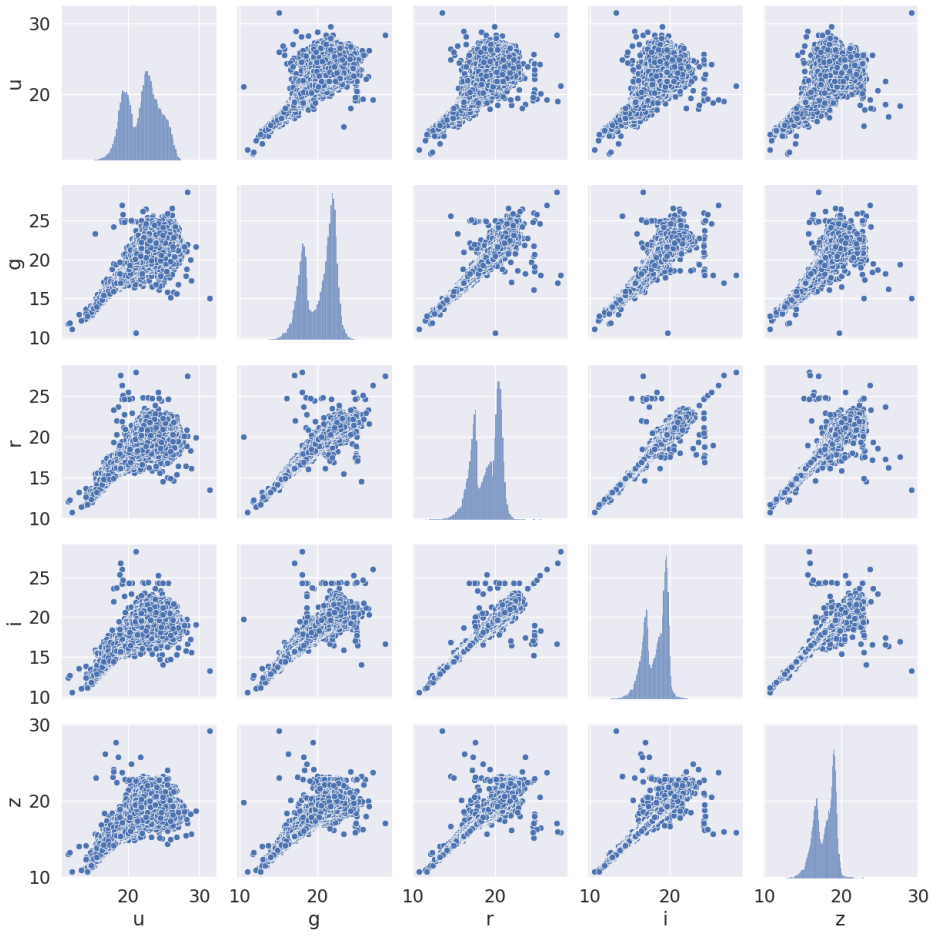

Figura 16: Bandas X Bandas (Happy)

Continuando com a análise com os dados dos erros das bandas na figura 17, percebe-se que não há uma relação aparente entre os erros e que os valores são distribuídos em uma ampla faixa.

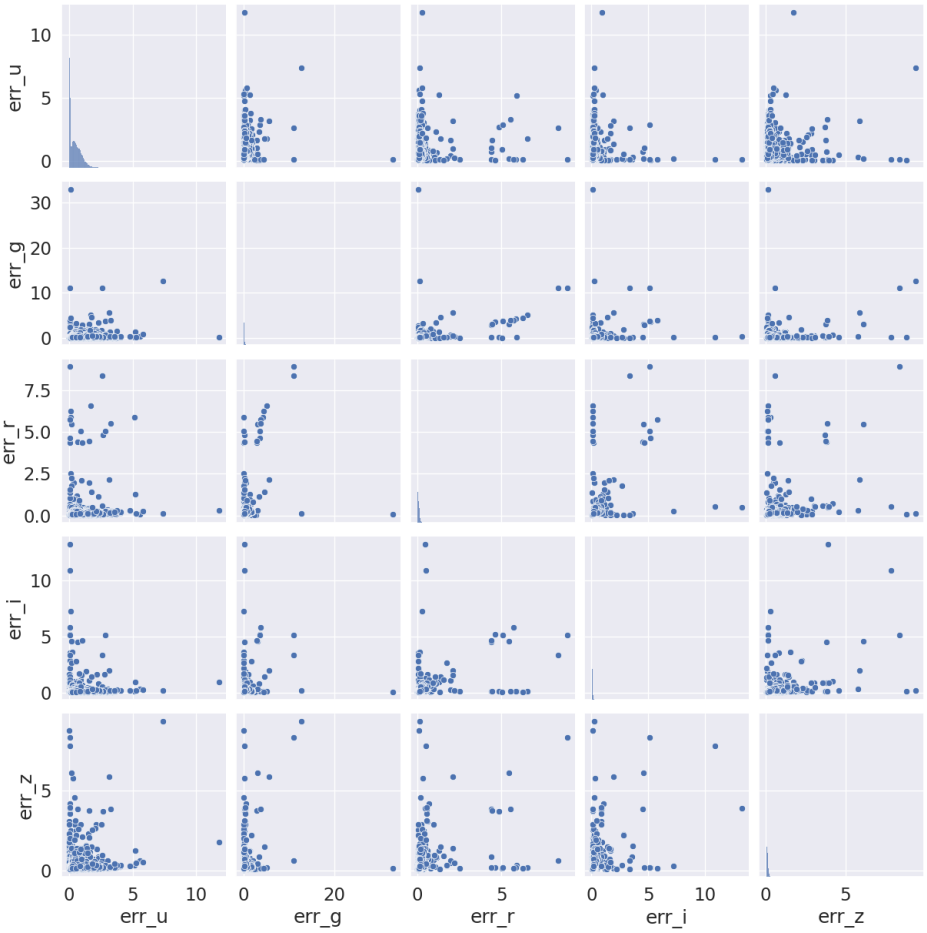

Figura 17: Erros X Erros (Happy)

A figura 18 expõe os dados das bandas com os erros. Observa-se uma leve tendência de crescimento junto com as magnitudes, para confirmar essa tendência em seguida será apresentado os gráficos com linhas de regressão.

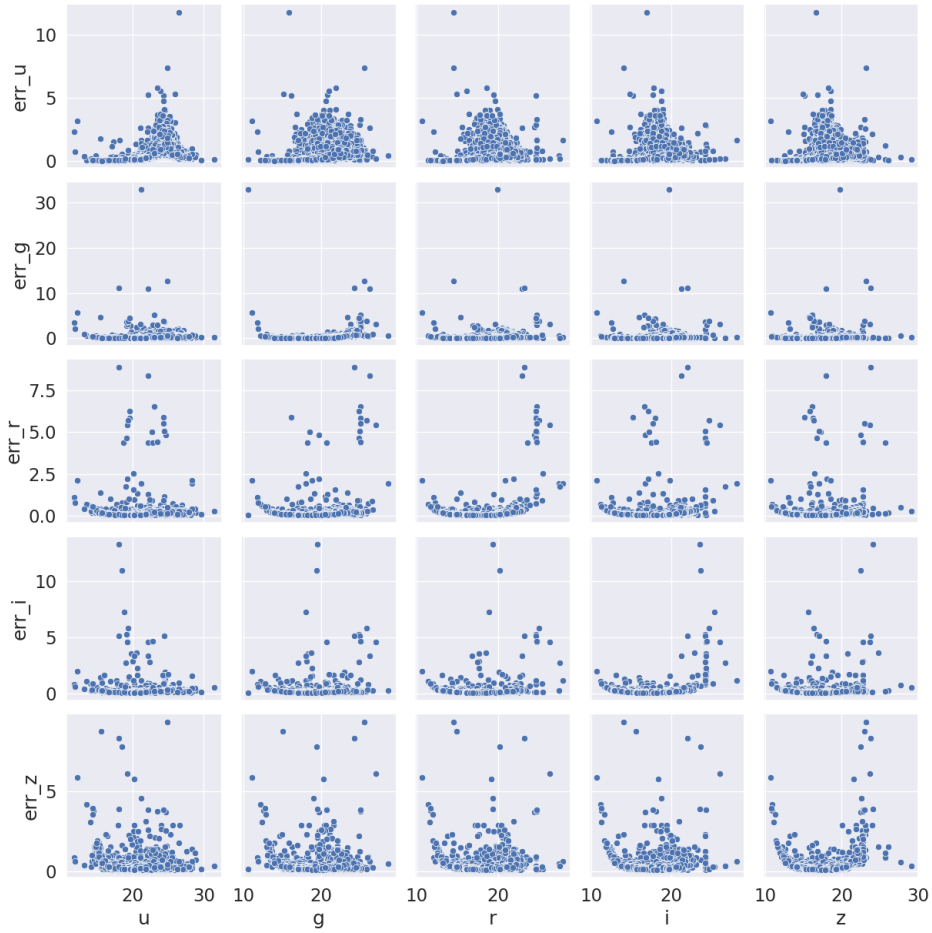

Figura 18: Bandas X Erros (Happy)

O gráfico na figura 19 mostra os valores da banda U pelos seus respectivos erros. Através dele pode-se observar que apesar a leve queda no final, em geral os valores dos erros aumentam com a magnitude.

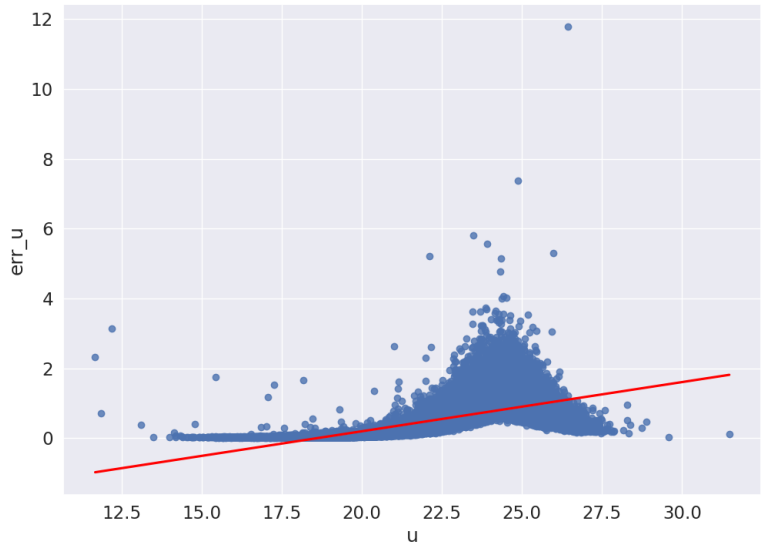

Figura 19: Banda U: Magnitude X Erro (Happy)

Seguindo para a próxima banda, a figura 20 expõe a relação entre o erro na banda G e a magnitude. Observa-se que o aumento do erro é bem menos expressivo mas ainda presente, preservando os padrões até então observados.

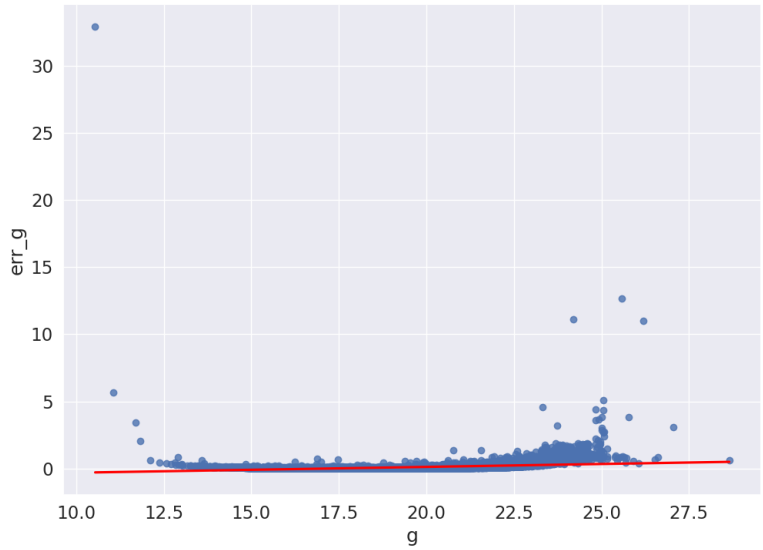

Figura 20: Banda G: Magnitude X Erro (Happy)

Analisando a banda R através do gráfico na figura 21, pode-se notar que ainda é mantido o padrão esperado de aumento dos erros, porém como na banda G esse aumento é inferior aos demais observados.

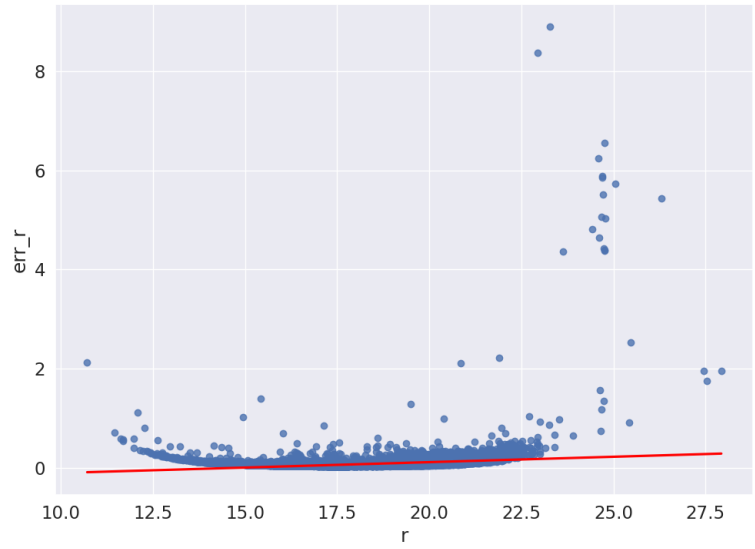

Figura 21: Banda R: Magnitude X Erro (Happy)

Partindo para banda I, o gráfico na figura 22. Continua expondo uma relação de aumento dos valores dos erros junto com os da banda e também fica notável a diferença entre os conjunto de dados Teddy e Happy, sendo o Happy o com valores em uma faixa bem mais ampla.

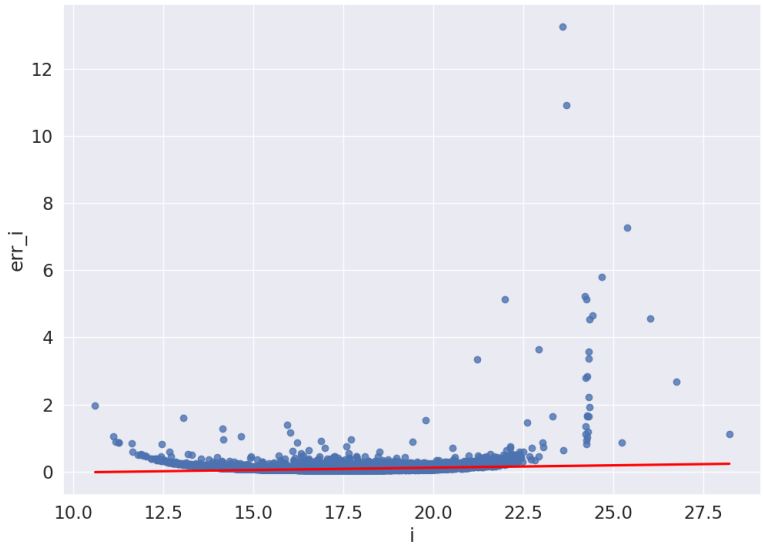

Figura 22: Banda I: Magnitude X Erro (Happy)

Analisando a última banda (Z) através do gráfico na imagem 23. Por fim, fica nítida (com base no Teddy e Happy) a relação de aumento de erros junto com os valroes de magnitudes e também o fato do conjunto de dados Happy conter valores de erros em uma faixa mais ampla.

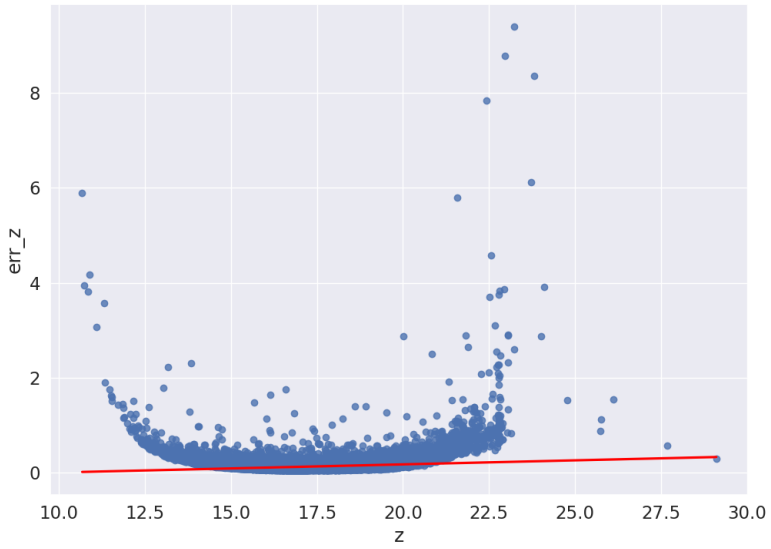

Figura 23: Banda Z: Magnitude X Erro (Happy)

4.3 Etapas dos experimentos

O código fonte dos experimentos pode ser encontrado na URL `https://github.com/MLRG-CEFET-RJ/rna-cdroput/tree/strategies`. Os conjuntos de dados utilizados nas próximas seções passaram pelas seguintes etapas durante os experimentos:

1. Download do conjunto de dados e definição das propriedades de interesse
2. Treinamento e predição de erros (acréscimo desse trabalho)
   (a) Carregamento do conjunto de dados
   (b) Aplicação da técnica de *k-fold cross-validation* para divisão do conjunto de dados
   (c) Aplicação da técnica de *grid-search* para otimização de hiperparâmetros
   (d) Treinamento dos modelos e seleção do melhor para predição
   (e) Predição de erros
3. Treinamento e predição de redshift (objetivo principal)
   (a) Carregamento do conjunto de dados com **com predição dos erros**
   (b) Divisão dos conjunto de dados em treino, teste e validação
   (c) *Scaling* dos valores para melhor treinamento do modelo

(d) Treinamento de predição de redshift (geração de modelos)

(e) Predição de redshift com melhor modelo gerado

No presente trabalho, foram utilizados os seguintes métodos para o passo prévio de mapeamento de magnitudes para erros:

1. Regressor Decision Tree com estratégia um para um (dt-1-x-1)
2. Regressor Decision Tree com estratégia muitos para um (dt-m-x-1)
3. Regressor Decision Tree com estratégia muitos para muitos (dt-m-x-m)
4. Regressor Isotonic com estratégia um para um (ir-1-x-1)
5. Regressor K Nearest Neighbors com estratégia um para um (knn-1-x-1)
6. Regressor K Nearest Neighbors com estratégia muitos para um (knn-m-x-1)
7. Regressor K Nearest Neighbors com estratégia muitos para muitos (knn-m-x-m)
8. Regressor Multilayer Perceptron com estratégia um para um (mlp-1-x-1)
9. Regressor Multilayer Perceptron com estratégia muitos para um (mlp-m-x-1)
10. Regressor Multilayer Perceptron com estratégia muitos para muitos (mlp-m-x-m)
11. Regressor Random Forest com estratégia um para um (rf-1-x-1)
12. Regressor Random Forest com estratégia muitos para um (rf-m-x-1)
13. Regressor Random Forest com estratégia muitos para muitos (rf-m-x-m)
14. Regressor XGBoost com estratégia um para um (xgb-1-x-1)
15. Regressor XGBoost com estratégia muitos para um (xgb-m-x-1)
16. Regressor XGBoost com estratégia muitos para muitos (xgb-m-x-m)

Primeiramente é realizado o download dos conjuntos de dados através de suas URLs e logo após é definido as propriedades de interesse desse trabalho como mostrado na tabela 6. Durante os experimentos, essa etapa foi executada apenas uma única vez para obter e transformar os dados.

Após o a primeira etapa, os conjuntos de dados são divididos em conjuntos de treino (80% das amostras) e de teste (20% das amostras), onde o conjunto de treino tem como finalidade treinar os modelos e o conjunto de teste será utilizado para mensurar a efetividade dos modelos. Após essa divisão será aplicada a técnica *k-fold cross-validation* no conjunto de teste, onde cada camada possuí 20% das amostras do conjunto de teste [1] como representado na Figura 24. Além da técnica de *k-fold cross-validation* também é aplicada a técnica de *grid-search* para busca e otimização dos hiperparâmetros. Vale ressaltar que cada modelo tem sua própria *grid* de hiperparâmetros fazendo essa etapa ser executada uma vez para cada modelo. Depois dessas configurações, é realizado o treinamento dos modelos e selecionado o melhor modelo pela métrica MSE para predição dos erros.

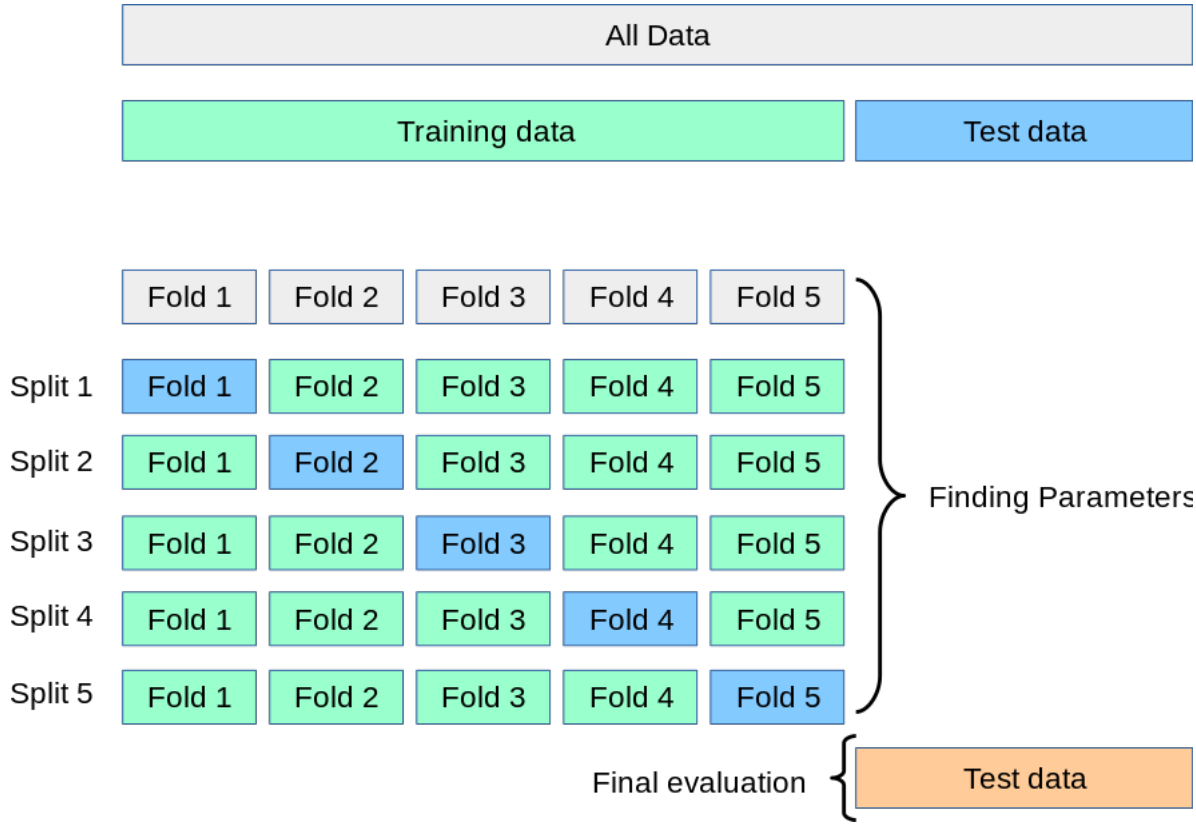

Figura 24: Validação cruzada de K camadas [Pedregosa et al., 2011]

Finalmente, é carregado o conjunto de dados com a predição dos erros para o treinamento e predição de redshift. Esses dados carregados inicialmente são separados em treino (60% das amostras), teste (20% das amostras) e validação (20% das amostras). A seguir, os dados são transformados para otimização do treinamento dos modelos utilizando o *StandardScaler* do *Scikit-Learn* (sklearn). Essa transformação é dada pela seguinte equação 4.1:

[1] K-fold cross-validation `https://scikit-learn.org/stable/modules/cross_validation.html`.

$$z = \frac{x - \mu}{\sigma} \tag{4.1}$$

Onde z é o resultado da transformação, x é o valor da observação, μ é a média da amostra e σ é o desvio padrão da amostra.

Após a transformação e treinamento, é selecionado o melhor modelo também pela métrica MSE e realizada predições com ele, mais a frente no texto está compilado o resultado de todos os modelos.

4.4 Resultados

Essa seção tem como objetivo apresentar os resultados dos experimentos executados, na Subseção 4.4.1 é documentado os resultados da predição dos erros das bandas e na Subseção 4.4.2 é exibido os resultados das predições de redshift.

4.4.1 Resultados das Predições dos Erros

Para o resultado, os experimentos foram executados com o banco de dados Teddy e Happy. Na Tabela 11 é organizado o resultado parcial dos modelos ordenado pelo maior MSE. Os melhores resultados foram obtidos pela estratégia xgb-m-x-m. Nos experimentos, é notável que a estratégia $m \times m$ e $m \times 1$ obtiveram resultados melhores que a estratégia 1×1 (única utilizada no trabalho de Fialho [2020]). Dito isso, é esperado que na Subseção 4.4.2 desse trabalho haja uma melhora na predição de redshift. Realizamos também, uma extensa busca para otimizar hiperparâmetros, os resultados da mesma estão localizadas na seguinte URL: `https://github.com/MLRG-CEFET-RJ/rna-cdroput/blob/strategies/docs/tcc_cassio_jorge_appendix.pdf`.

Tabela 11: Predição de erros nos modelos de regressão.

mse	dataset	regressor	estratégia	mae	r2
0.002846	teddy_data	xgb	m_x_1	0.024119	0.532173
0.002846	teddy_data	xgb	m_x_m	0.024119	0.532173
0.002942	teddy_data	rf	m_x_m	0.024728	0.512162
0.002948	teddy_data	rf	m_x_1	0.024648	0.524151
0.003112	teddy_data	dt	m_x_1	0.025498	0.481204
0.003114	teddy_data	dt	m_x_m	0.025324	0.475760
0.003181	teddy_data	ir	1_x_1	0.025956	0.457364
0.003185	teddy_data	dt	1_x_1	0.025927	0.460803
0.003189	teddy_data	dt	1_x_1	0.025947	0.460586
0.003203	teddy_data	knn	m_x_m	0.025488	0.472323
0.003209	teddy_data	knn	m_x_1	0.025518	0.472204
0.003215	teddy_data	xgb	1_x_1	0.026044	0.460414
0.003281	teddy_data	mlp	m_x_1	0.028635	0.260755
0.003300	teddy_data	mlp	1_x_1	0.027112	0.402090
0.003467	teddy_data	mlp	m_x_m	0.029392	0.327985
0.003476	teddy_data	knn	1_x_1	0.027251	0.408468
0.004576	teddy_data	rf	1_x_1	0.031091	0.221595
0.006322	teddy_data	dt	1_x_1	0.036322	-0.072767
0.010378	happy_data	xgb	m_x_m	0.036729	0.737010
0.011196	happy_data	xgb	m_x_1	0.037426	0.642801
0.011488	happy_data	rf	m_x_1	0.036675	0.716264
0.011560	happy_data	rf	m_x_m	0.037155	0.691945
0.011951	happy_data	knn	m_x_m	0.039113	0.695380
0.011962	happy_data	knn	m_x_1	0.038435	0.693278
0.013925	happy_data	mlp	m_x_1	0.046036	0.597806
0.015617	happy_data	dt	m_x_m	0.040491	0.504403
0.016021	happy_data	dt	m_x_1	0.040866	0.483414
0.017125	happy_data	xgb	1_x_1	0.045190	0.429715
0.017127	happy_data	knn	1_x_1	0.046564	0.524801
0.017433	happy_data	mlp	1_x_1	0.048540	0.450256
0.018734	happy_data	dt	1_x_1	0.045279	0.262437
0.019322	happy_data	ir	1_x_1	0.049218	0.559157
0.022097	happy_data	rf	1_x_1	0.052604	0.337959
0.026801	happy_data	mlp	m_x_m	0.066316	0.390566

Como definido no Seção 3.1, para a estratégia $m \times m$ não há necessidade de múltiplos regressores, já nas demais estratégias, são utilizados cinco regressores, um para cada banda. A Tabela 12 apresenta uma visão geral dos hiperparâmetros utilizados nos modelos.

Tabela 12: Configuração de Hiperparâmetros

técnica de regressão	hiperparâmetro	descrição	valores			
Decision Tree (dt)	max_depth	profundidade da Árvore	5	10	15	
	criterion	função avaliadora da qualidade da divisão	squared_error	friedman_mse	absolute_error	poisson
	splitter	estratégia para escolher a divisão em cada nó	best	random		
Isotonic Regressor (ir)	out_of_bounds	manipula valores X fora do domínio	clip			
Random Forest (rf)	criterion	função avaliadora da qualidade da divisão	squared_error	absolute_error	poisson	
Multilayer Perceptron (mlp)	hidden_layer_sizes	tamanho das camadas ocultas	5	10	(5,5)	
	max_iter	máximo número de épocas	100	500		
KNeighbors Regressor (knn)	n_neighbors	número de vizinhos	4	5	6	
	weights	pesos	uniform	distance		
	algorithm	algorítimo utilizado para computar os vizinhos mais próximos	ball_tree	kd_tree	brute	
	p	paramêtro exponencial para métrica de Minkowski	1	2	3	
XGBoost (xgb)	max_depth	tamanho das árvores	1	5	10	
	objective	objetivo da aprendizagem	reg:squarederror			

4.4.2 Resultados das predições de desvio fotométrico

Abaixo a tabela 13 exibe todos os resultados de predição de redshift com os modelos de regressão gerados a partir da aplicação das estratégias. Para isso, foi pego o resultado da predição dos erros e treinado as redes neurais preditoras de redshift com 3000 épocas.

Tabela 13: Predição de Redshifts com os Modelos de Regressão

model	MSE_MEAN	RMSE_MEAN	MAD_MEAN	R2_MEAN
teddy_ir_1_x_1	0.091410	0.302180	0.179227	-4.940906
teddy_dt_1_x_1	0.105769	0.324784	0.192402	-6.950980
teddy_dt_m_x_m	0.112741	0.335727	0.197169	-6.453313
teddy_xgb_m_x_m	0.113859	0.335369	0.197210	-8.374302
teddy_mlp_m_x_1	0.118965	0.335773	0.196174	-10.512154
teddy_rf_m_x_1	0.119302	0.345232	0.201587	-6.621311
teddy_rf_1_x_1	0.122457	0.348428	0.202346	-8.770517
teddy_xgb_m_x_1	0.123185	0.350472	0.206192	-6.496686
teddy_knn_m_x_m	0.126782	0.333655	0.189607	-2.210034
teddy_mlp_m_x_m	0.126839	0.356144	0.209069	-7.607231
teddy_mlp_1_x_1	0.168464	0.410005	0.233824	-11.245216
teddy_rf_m_x_m	0.198311	0.440281	0.246277	-15.948913
teddy_xgb_1_x_1	0.203625	0.450003	0.251888	-14.430828
teddy_knn_1_x_1	0.214463	0.456586	0.251655	-17.746222
teddy_dt_m_x_1	0.233239	0.459060	0.249354	-5.450222
teddy_knn_m_x_1	0.278852	0.527160	0.282784	-15.165592
happy_mlp_m_x_m	1.619166	1.270450	10.868624	-14.963033
happy_rf_m_x_1	2.169234	1.275063	1.848396	-0.769197
happy_knn_m_x_1	10.311319	3.114841	1.986110	-169.522443
happy_rf_m_x_m	10.787219	3.155169	2.035841	-185.333183
happy_ir_1_x_1	12.754635	3.144689	4.619334	-264.869847
happy_knn_m_x_m	12.844561	3.343460	2.195266	-241.888155
happy_mlp_m_x_1	27.540302	4.549642	2.298342	-582.134326
happy_xgb_m_x_m	31.118286	5.540914	1.335114	-423.228498
happy_dt_m_x_1	35.915131	5.266885	1.760643	-748.391618
happy_mlp_1_x_1	41.725721	6.455882	1.270128	-420.307282
happy_knn_1_x_1	42.265273	6.063922	1.393068	-121.701427
happy_dt_m_x_m	53.740124	7.325635	1.230697	-536.712560
happy_dt_1_x_1	59.923191	7.400239	1.266176	-252.201375
happy_rf_1_x_1	81.847478	6.744278	1.051960	-1857.106640
happy_xgb_1_x_1	89.152050	6.801282	0.629104	-2028.739645
happy_xgb_m_x_1	100.853628	9.938563	1.164804	-756.698360

Capítulo 5

Considerações finais

Esse capítulo tem como objetivo apresentar as considerações finais que engloba esse trabalho. Na Seção 5.1, apresentamos uma análise retrospectiva do trabalho realizado. Na Seção 5.2, apresentamos temas possíveis para trabalhos futuros.

5.1 Análise retrospectiva

Em comparação ao início do trabalho, expandimos os experimentos relativos aos mapeadores de magnitudes para erros, explorando mais algoritmos de aprendizado, implementando e incluindo os mesmos ao trabalho. Utilizamos os conjuntos de dados Teddy e Happy, aprimorando o conhecimento das bases de dados na etapa de pré-processamento, no entanto não foi possível chegar a um resultado final com o SDSS devido ao tempo de processamento do modelo e ajuste analítico de informações finais. Além disso, foi aprimorada e encorporada a busca de hiperparâmetros nos modelos de predição de erros visando encontrar os melhores.

Na completude do trabalho foi possível analisar o desempenho dos modelos de erros nas diferentes estratégias implementadas e utilizado nos modelos de predição de desvios fotométricos (redshifts).

Comparando os resultados apresentados na tabela 13chegamos a conclusão que para o conjunto de dados Teddy as diferentes estratégias não obtiveram resultados melhores, porém eles foram bem próximos. Já para o conjunto de dados Happy as diferentes estratégias conseguiram resultados melhores, colocando a solução original em quinto lugar.

5.2 Trabalhos futuros

Na Seção 2.1.4 mencionamos a existência de vários sistemas de magnitude usados na astronomia os quais possuem diferentes conjuntos de bandas. Uma possível continuidade do trabalho, seria utilizar um sistema de magnitudes diferente do ugriz, o qual utilizamos no trabalho.

Para incrementar esse trabalho, podemos executar a rede neural de predição de redshifts com um maior número de execuções, pois cada rede neural aprende de forma aleatória, podendo desta forma gerar resultados ainda não alcançados ou previstos.

Além disso, vale lembrar que dado a enorme quantidade de dados do dataset SDSS, o mesmo não foi trabalhado. Ele é importante pois os modelos gerados podem ser validados para uma amostra com muitas observações. Logo, implementar a utilização do SDSS com diferentes tamanhos de conjuntos de treinamento pode ser uma possibilidade, visando entender se a rede se beneficiará do conjunto de maior volume de dados.

Referências Bibliográficas

Abdurro'uf, Accetta, K., Aerts, C., Aguirre, V. S., Ahumada, R., Ajgaonkar, N., Ak, N. F., Alam, S., Prieto, C. A., Almeida, A., Anders, F., Anderson, S. F., Andrews, B. H., Anguiano, B., Aquino-Ortiz, E., Aragon-Salamanca, A., Argudo-Fernandez, M., Ata, M., Aubert, M., Avila-Reese, V., Badenes, C., Barba, R. H., Barger, K., Barrera-Ballesteros, J. K., Beaton, R. L., Beers, T. C., Belfiore, F., Bender, C. F., Bernardi, M., Bershady, M. A., Beutler, F., Bidin, C. M., Bird, J. C., Bizyaev, D., Blanc, G. A., Blanton, M. R., Boardman, N. F., Bolton, A. S., Boquien, M., Borissova, J., Bovy, J., Brandt, W. N., Brown, J., Brownstein, J. R., Brusa, M., Buchner, J., Bundy, K., Burchett, J. N., Bureau, M., Burgasser, A., Cabang, T. K., Campbell, S., Cappellari, M., Carlberg, J. K., Wanderley, F. C., Carrera, R., Cash, J., Chen, Y.-P., Chen, W.-H., Cherinka, B., Chiappini, C., Choi, P. D., Chojnowski, S. D., Chung, H., Clerc, N., Cohen, R. E., Comerford, J. M., Comparat, J., da Costa, L., Covey, K., Crane, J. D., Cruz-Gonzalez, I., Culhane, C., Cunha, K., Dai, Y. S., Damke, G., Darling, J., au2, J. W. D. J., Davies, R., Dawson, K., Lee, N. D., Diamond-Stanic, A. M., Cano-Diaz, M., Sanchez, H. D., Donor, J., Duckworth, C., Dwelly, T., Eisenstein, D. J., Elsworth, Y. P., Emsellem, E., Eracleous, M., Escoffier, S., Fan, X., Farr, E., Feng, S., Fernandez-Trincado, J. G., Feuillet, D., Filipp, A., Fillingham, S. P., Frinchaboy, P. M., Fromenteau, S., Galbany, L., Garcia, R. A., Garcia-Hernandez, D. A., Ge, J., Geisler, D., Gelfand, J., Geron, T., Gibson, B. J., Goddy, J., Godoy-Rivera, D., Grabowski, K., Green, P. J., Greener, M., Grier, C. J., Griffith, E., Guo, H., Guy, J., Hadjara, M., Harding, P., Hasselquist, S., Hayes, C. R., Hearty, F., Hernandez, J., Hill, L., Hogg, D. W., Holtzman, J. A., Horta, D., Hsieh, B.-C., Hsu, C.-H., Hsu, Y.-H., Huber, D., Huertas-Company, M., Hutchinson, B., Hwang, H. S., Ibarra-Medel, H. J., Chitham, J. I., Ilha, G. S., Imig, J., Jaekle, W., Jayasinghe, T., Ji, X., Johnson, J. A., Jones, A., Jonsson, H., Katkov, I., Khalatyan, D. A., Kinemuchi, K., Kisku, S., Knapen, J. H., Kneib, J.-P., Kollmeier, J. A., Kong, M., Kounkel, M., Kreckel, K., Krishnarao, D., Lacerna, I., Lane, R. R., Langgin, R., Lavender, R., Law, D. R., Lazarz, D., Leung, H. W., Leung, H.-H., Lewis, H. M., Li, C., Li, R., Lian, J., Liang, F.-H., Lin, L., Lin, Y.-T., Lin, S., Lintott, C., Long, D., Longa-Pena, P., Lopez-Coba, C., Lu, S., Lundgren, B. F., Luo, Y., Mackereth, J. T., de la Macorra, A., Mahadevan, S., Majewski, S. R., Manchado, A., Mandeville, T., Maraston, C., Margalef-Bentabol, B., Masseron, T., Masters, K. L., Mathur, S., McDermid, R. M., Mckay, M., Merloni, A., Merrifield, M., Meszaros, S., Miglio, A., Mille, F. D., Minniti,

D., Minsley, R., Monachesi, A., Moon, J., Mosser, B., Mulchaey, J., Muna, D., Munoz, R. R., Myers, A. D., Myers, N., Nadathur, S., Nair, P., Nandra, K., Neumann, J., Newman, J. A., Nidever, D. L., Nikakhtar, F., Nitschelm, C., O'Connell, J. E., Garma-Oehmichen, L., de Oliveira, G. L. S., Olney, R., Oravetz, D., Ortigoza-Urdaneta, M., Osorio, Y., Otter, J., Pace, Z. J., Padilla, N., Pan, K., Pan, H.-A., Parikh, T., Parker, J., Peirani, S., Ramirez, K. P., Penny, S., Percival, W. J., Perez-Fournon, I., Pinsonneault, M., Poidevin, F., Poovelil, V. J., Price-Whelan, A. M., de Andrade Queiroz, A. B., Raddick, M. J., Ray, A., Rembold, S. B., Riddle, N., Riffel, R. A., Riffel, R., Rix, H.-W., Robin, A. C., Rodriguez-Puebla, A., Roman-Lopes, A., Roman-Zuniga, C., Rose, B., Ross, A. J., Rossi, G., Rubin, K. H. R., Salvato, M., Sanchez, S. F., Sanchez-Gallego, J. R., Sanderson, R., Rojas, F. A. S., Sarceno, E., Sarmiento, R., Sayres, C., Sazonova, E., Schaefer, A. L., Schiavon, R., Schlegel, D. J., Schneider, D. P., Schultheis, M., Schwope, A., Serenelli, A., Serna, J., Shao, Z., Shapiro, G., Sharma, A., Shen, Y., Shetrone, M., Shu, Y., Simon, J. D., Skrutskie, M. F., Smethurst, R., Smith, V., Sobeck, J., Spoo, T., Sprague, D., Stark, D. V., Stassun, K. G., Steinmetz, M., Stello, D., Stone-Martinez, A., Storchi-Bergmann, T., Stringfellow, G. S., Stutz, A., Su, Y.-C., Taghizadeh-Popp, M., Talbot, M. S., Tayar, J., Telles, E., Teske, J., Thakar, A., Theissen, C., Thomas, D., Tkachenko, A., Tojeiro, R., Toledo, H. H., Troup, N. W., Trump, J. R., Trussler, J., Turner, J., Tuttle, S., Unda-Sanzana, E., Vazquez-Mata, J. A., Valentini, M., Valenzuela, O., Vargas-Gonzalez, J., Vargas-Magana, M., Alfaro, P. V., Villanova, S., Vincenzo, F., Wake, D., Warfield, J. T., Washington, J. D., Weaver, B. A., Weijmans, A.-M., Weinberg, D. H., Weiss, A., Westfall, K. B., Wild, V., Wilde, M. C., Wilson, J. C., Wilson, R. F., Wilson, M., Wolf, J., Wood-Vasey, W. M., Yan, R., Zamora, O., Zasowski, G., Zhang, K., Zhao, C., Zheng, Z., Zheng, Z., and Zhu, K. (2022). The seventeenth data release of the sloan digital sky surveys: Complete release of manga, mastar and apogee-2 data. 12

Azad, M., Chikalov, I., Hussain, S., and Moshkov, M. (2021). Entropy-based greedy algorithm for decision trees using hypotheses. *Entropy*, 23(7). 13

Barlow, R. E. (1972). Statistical inference under order restrictions; the theory and application of isotonic regression. Technical report. 14

Beck, R., Lin, C.-A., Ishida, E. E. O., Gieseke, F., de Souza, R. S., Costa-Duarte, M. V., Hattab, M. W., and and, A. K.-M. (2017). On the realistic validation of photometric redshifts. *Monthly Notices of the Royal Astronomical Society*, 468(4):4323–4339. 26

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM. 14, 25

Chen, X. and Ishwaran, H. (2012). Random forests for genomic data analysis. *Genomics*, 99(6):323–329. 14

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27. 14

DeGraaff, R. B., Blakeslee, J. P., Meurer, G. R., and Putman, M. E. (2007). A galaxy in transition: Structure, globular clusters, and distance of the star-forming s0 galaxy NGC 1533 in dorado. *The Astrophysical Journal*, 671(2):1624–1639. 7

D'Isanto, A. and Polsterer, K. L. (2018). Photometric redshift estimation via deep learning. *Astronomy & Astrophysics*, 609:A111. 3, 18

Fialho, R. C. S. (2020). Estimando redshifts fotomÉtricos com regularizaÇÃo sensÍvel aos erros. Mestrado em ciência da computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Santa Maria. iii, iv, 2, 3, 4, 5, 6, 9, 10, 15, 16, 17, 19, 20, 23, 43

Fletcher, S. and Islam, M. Z. (2019). Decision tree classification with differential privacy: A survey. *ACM Computing Surveys*, 52(4). 13

Fukunaga, K. and Hostetler, L. (1975). k-nearest-neighbor bayes-risk estimation. *IEEE Transactions on Information Theory*, 21(3):285–293. 14

Gerdes, D. W., Sypniewski, A. J., McKay, T. A., Hao, J., Weis, M. R., Wechsler, R. H., and Busha, M. T. (2010). ArborZ: Photometric redshifts using boosted decision trees. *Astrophysical Journal*, 715(2):823–832. 13

Guresen, E. and Kayakutlu, G. (2011). Definition of Artificial Neural Networks with comparison to other networks. *Procedia Computer Science*, 3(May):426–433. 12

Guresen, E., Kayakutlu, G., and Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, 38(8):10389–10397. 13

Hellman, M. E. (1970). The nearest neighbor classification rule with a reject option. *IEEE Transactions on Systems Science and Cybernetics*, 6(3):179–185. 14

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674. 13

Hoyle, B. (2016). Measuring photometric redshifts using galaxy images and deep neural networks. *Astronomy and Computing*, 16:34–40. 18

Hubble, E. (1929). A relation between distance and radial velocity among extra-galactic nebulae. *Proceedings of the National Academy of Sciences*, 15(3):168–173. 1

Inocente, G., Garbuglio, D. D., and Ruas, P. M. (2022). Multilayer perceptron applied to genotypes classification in diallel studies. *Scientia Agricola*, 79(3). 12

Kaufmann, W. J. and Comins, N. F. (2014). *Discovering the Universe*. W. H. Freeman and Company. 1, 6, 7, 8, 9

Kremer, J., Stensbo-Smidt, K., Gieseke, F., Pedersen, K. S., and Igel, C. (2017). Big universe, big data: Machine learning and image analysis for astronomy. *IEEE intelligent systems*, 32(2):16–22. 1

Mickaelian, A. M. (2016). Astronomical surveys and big data. *Open Astronomy*, 25(1). 11

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York. 2

Mukhopadhyay, S. (2003). Neural and adaptive systems: fundamentals through simulations: José c. principe, neil r. euliano and w. curt lefebvre; john wiley & sons, inc., usa, 2000, isbn 0-471-35167-9. *Automatica*, 39(7):1313–1315. 13

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. vii, 25, 42

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883. revision #137311. 13, 14

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958. 3

Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA. 25