UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

SOFIA GARCIA TELLES BRITO

# Estimativa do redshift fotométrico de galáxias utilizando redes neurais artificiais

Campinas
11/05/2024

SOFIA GARCIA TELLES BRITO

# Estimativa do redshift fotométrico de galáxias utilizando redes neurais artificiais*

Monografia apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos para obtenção de créditos na disciplina Projeto de Extensão Supervisionado, sob a orientação do(a) Profª. Flávia Sobreira.

*Este trabalho foi financiado pelo PIBIC, projeto 2023/2024.

# Resumo

Esse trabalho consiste na revisão do *paper ANNz2 - Photometric Redshift and Probability Distribution Function Estimation Using Machine Learning* from SADEH, I., ABDALLA, F. B., LAHAV. O *ANNz2* é uma ferramenta que utiliza Redes Neurais Artificiais no treinamento de um algoritmo para que este consiga fazer estimativas do *redshift* fotométrico, $z_{photo}$, muito próximas do *redshift* espectroscópico, $z_{spec}$. Foram coletados os dados das magnitudes $m_u$, $m_g$, $m_r$, $m_i$ e $m_z$, assim como seus respectivos erros associados, e dos *redshifts* espectroscópicos do catálogo do *Sloan Digital Sky Survey* (SDSS), *Release* 16 (DR16) a fim de testar dois modos de operação do *ANNz2 Single Regression* e *Random Classification*.

# Abstract

This work consists in reviewing the paper ANNz2 - Photometric Redshift and Probability Distribution Function Estimation Using Machine Learning, from SADEH, I., ABDALLA, F. B., LAHAV. ANNz2 is a tool that uses Artificial Neural Networks to train an algorithm so that it can estimate the photometric redshift, $z_{photo}$, very close to the spectroscopic redshift, $z_{spec}$. Data on magnitudes $m_u$, $m_g$, $m_r$, $m_i$ e $m_z$, as well as their respective associated errors, were found on the Sloan Digital Sky Survey (SDSS), Release 16 (DR16), catalog in order to test two operating modes of ANNz2: Single Regression and Random Classification.

# Conteúdo

# 1 Introdução

A observação do céu pode nos ajudar a entender o comportamento do universo desde os seus primórdios. Desde a descoberta da expansão cósmica em 1920 por Edwin Hubble, a cosmologia se tornou uma ciência de *big data* com a construção de telescópios modernos capazes de observar galáxias cada vez mais distantes. Uma informação importante é extraída do espectro da radiação destas galáxias. Quando se compara as linhas de absorção ou emissão da radiação destes objetos com as de vários compostos químicos na Terra, percebe-se um aumento no comprimento de onda dos fotóns caracterizado por um desvio na direção do vermelho, o que pode ser melhor visualizado com a Figura 1. Isso indica que as galáxias estão se afastando de nós [1].

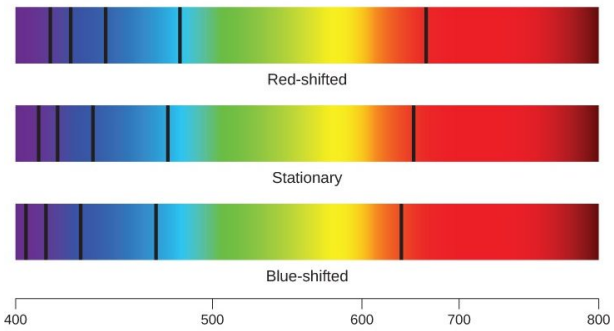


Figura 1: Linhas espectrais com desvio para o vermelho (*red-shifted*), sem desvio (*stationary*) e com desvio para o azul (*blue-shifted*). (Fonte: https://noic.com.br/astronomia/curso/miscelanea/redshift-e-lei-de-hubble/)

O astrônomo americano Vesto Melvin Slipher, do Observatório Lowell, em 1912, ao analisar as linhas espectrais de 41 galáxias, percebeu que a maioria apresentava deslocamento espectral para o vermelho. Objetivamente, o *redshift* é a medida do aumento no comprimento de onda $\lambda$, ou diminuição da frequência $\nu$ (já que $\lambda = \frac{c}{\nu}$) da onda eletromagnética, o que ocorre quando a fonte emissora se afasta do observador.

*Redshifts* espectroscópicos são estimados utilizando as linhas espectrais de objetos observados. Isto é possível em dados de telescópios como o Sloan Digital Sky Survey (SDSS) [5] e Dark Energy Spectroscopic Instrument (DESI) [6] que observam o espectro inteiro de uma galáxia. Já telescópios como o Dark Energy Survey (DES) [7] e o Large Synoptic Survey Telescopic (LSST) [8] mapeiam o céu usando imagens de bandas espectrais. Neste caso, o *redshift* é estimado com base nas cores das galáxias em três ou mais

filtros e também em outras propriedades que podem ser obtidas das imagens, como o tamanho angular ou o índice de concentração [3].

Usando dados de telescópios espectroscópicos, podemos estimar os *redshifts* dos objetos com muita precisão. Porém, para observar o espectro inteiro de uma galáxia, é necessário um tempo de observação grande. Já telescópios fotométricos, embora percam a precisão na estimativa dos *redshifts*, conseguem observar uma quantidade muito maior de galáxias, aumentando a precisão da análise estatística feita com os dados. Se sabemos modelar o erro na estimativa de *redshift* fotométrico, então é vantajoso usar estes dados.

Nesse projeto, o código ANNz2 do *paper ANNz2 - Photometric Redshift and Probability Distribution Function Estimation Using Machine Learning* [4] foi escolhido como ferramenta para fazer estimativas do *redshift* fotométrico utilizando redes neurais e diversos métodos de *Machine Learning*. Para isso, foram utilizados os dados das magnitudes e dos *redshifts* espectroscópicos presentes na base de dados do SDSS (*Sloan Digital Sky Survey*), a fim de utilizar esses dados para, com base no que foi feito no *paper* [4], obter resultados utilizando os diversos métodos de treinamento nele apresentados.

# 2 Desenvolvimento

Em primeira análise, é necessário explicar o que é *Machine Learning* e quais são os métodos de treinamento existentes nesse tópico.

## 2.1 Definição de Machine Learning

Uma definição de *Machine Learning* pode ser encontrada na referência [13] (GÉRON, 2017):

"*Machine Learning* é a ciência (ou a arte) de programar computadores, de maneira que eles consigam aprender através dos dados."

Na mesma referência [13] (GÉRON, 2017), também se encontram duas definições mais gerais de *Machine Learning*:

"*Machine Learning* é o campo de estudo que dá aos computadores a habilidade de aprender sem ter sido explicitamente programado."(Arthur Samuel, 1959).

"Se diz que um programa computacional aprende de uma experiência E, com

respeito a uma tarefa T com uma performance P, se sua performance P em T aumenta com a experiência E."(Tom Mitchel, 1997).

Uma outra definição támbém é encontrada na referência [11]: *Machine Learning* se refere a um conjunto de técnicas para interpretação de dados em que se compara esses dados a modelos para o comportamento dos dados. Alguns exemplos dessas técnicas são métodos de regressão, métodos de classificação supervisionada, *maximum likelihood estimators* e o método Bayesiano [11].

## 2.2 Alguns conceitos de Machine Learning

### 2.2.1 Performance

Performance, no contexto desse trabalho, é a habilidade de prever o *redshift* de uma galáxia individual de forma precisa, ou seja, com uma pequena incerteza quando comparado com o verdadeiro *redshift*, que aqui escolheremos como sendo o *redshift* espectroscópico [2].

### 2.2.2 Caracterização

Caracterização, no contexto desse trabalho, é a habilidade de abarcar as propriedades da distribuição de um conjunto de galáxias [2].

### 2.2.3 Função Densidade de Probabilidade (PDF)

A Função Densidade de Probabilidade (PDF), $h(x)$, quantifica a probabilidade de que um valor esteja entre $x$ e $x + dx$, que é igual a $h(x)dx$ [11]. A PDF do *redshift*, $p(z)$, nos dá a melhor representação do resultado de um algoritmo de *redshift* fotométrico [2]. A integral da PDF é chama Função Distribuição Cumulativa e é dada por [11]

$$H(x) = \int_{-\infty}^{x} h(x')dx'.$$

## 2.3 Algumas observações sobre métodos de treinamento em Machine Learning

Os métodos de *Machine Learning* podem ser distinguidos por alguns fatores: pelo conjunto de treinamento de galáxias e informações que vão ser usadas sobre as galáxias para prever o *redshift* (exemplos: cores de galáxias, magnitude do fluxo); pelo quanto eles são treinados para otimizar; e por outras suposições e escolhas que afetam a estimativa da quantidade alvo (exemplos: alguns métodos dividem o conjunto de treinamento em subconjuntos por propriedades distinguíveis e outros definem uma vizinhança usando galáxias como referência para estimar o *redshift* de um objeto)[2].
Existem alguma limitações possíveis se tratando de *Machine learning* e é muito importante estar vigilante quanto a essas limitações. Elas são: conjuntos de treinamento não-representativos para o conjunto alvo; conjuntos de treinamento contendo erros (nos *redshifts* ou nos observáveis fotométricos) e conjuntos de treinamento pouco diversos, enviesando o aprendizado [2].

## 2.4 Regressão

A Regressão é uma das técnicas utilizadas em *Machine Learning*. Ela se trata da relação entre variáveis dependentes, $y$, e um conjunto de variáveis independentes, $x$, que descreve o valor esperado de $y$ dado $x$: $E[y|x]$. [11]

No caso em que temos um modelo para a distribuição condicional com parâmetros $\theta$, escrevemos a função $y = f(x|\theta)$, sendo $y$ variável dependente e $x$ vetor independente. [11].

No caso da regressão linear, temos

$$y_i = \theta_0 + \theta_1 x_i + \epsilon_i,$$

onde o índice $i$ designa diferentes observações, $\theta_0$ e $\theta_1$ são os coeficientes que descrevem a função de regressão que estamos tentanto estimar e $\epsilon_i$ representa um termo extra de ruído. [11]

## 2.5 Classificação

Na classificação supervisionada, criamos classes de dados e agrupanmos esses dados em categorias de acordo com uma determinada propriedade. Relaciona-se, então, um conjunto de parâmetros com os conjuntos de classes já predefinidos [11].

## 2.6 ANNz2

No ANNz2, o conjunto de dados original utilizado (do SDSS - *Sloan Digital Sky Survey*) é separado em três partes: treinamento, validação e testagem. O conjunto de treinamento é usado para derivar o mapeamento entre *inputs* e *outputs*, enquanto que, a cada etapa do treinamento, o conjunto de validação é utilizado para estimar a convergência da solução comparando o resultado da estimativa com o valor do *output*. Já o conjunto de testagem é usado após o treinamento para analisar a performance deste. [4]

Os dados de *input* coletados do SDSS são as magnitudes $m_u$, $m_g$, $m_r$, $m_i$, $m_z$. O dado de *output* é a estimativa feita do *redshift* fotométrico [4]. Se olharmos para a Seção 2.3, que trata sobre Regressão, os *inputs* das magnitudes seriam as componentes do vetor $\mathbf{x}$ e o *output* seria o $y$. O valor do *redshift* espectroscópico serve como valor real do *redshift*, para comparar com o valor de *output* encontrado para o *redshift* fotométrico.

No ANNz2, os métodos de *Machine Learning* utilizados estão implementados no pacote TMVA do ROOT. O ROOT é um *framework* de processamento de dados do CERN (*European Organization for Nuclear Research*. O TMVA se trata de uma biblioteca do ROOT que contém diversas implementações de técnicas de *Machine Learning*, como *Neural Networks*, *Deep Networks*, *Multilayer Perceptron*, *Boosted/Bagged Decision Trees*, *Support Vector Machines*(CVM) e outros [12]. Os métodos que foram considerados mais adequados foram redes neurais artificais e *boosted decision trees* [4].

## 2.7 Redes neurais artificiais (ANNs)

Trata-se de um mapeamento entre o conjunto de variáveis de *input* (como magnitudes e cores) a uma ou mais varíáveis de output, que é feito calculando a soma com pesos da coleção de funções de resposta (*response functions*). As variáveis de *input*, as funções de resposta e as variáveis de *output* são chamadas de neurônios [4]. No ANNz2

a ANNs utilizada foi a *Multilayer Perceptron*. Nessa rede, os neurônios são organizados em pelo menos 3 camadas: *input*, ocultos e *output*. Uma representação esquemática dessa rede se encontra na figura 2 [4].

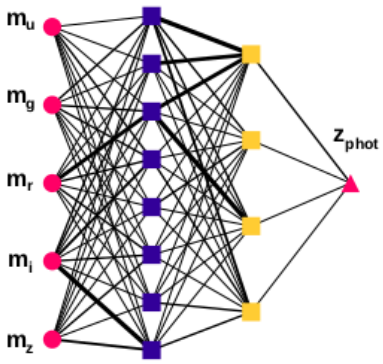


Figura 2: Representação esquemática de uma rede neural aritficial, com os neurônios representados por círculos (*inputs*), quadrados (ocultos) e triângulos (*outputs*). Referência: "*ANNz2 - Photometric redshift and probability distribution function estimation using machine learning*"[4].

O aprendizado ocorre pela mudança de pesos inter-neuroniais após cada elemento do conjunto de dados ter sido processado, usando um algoritmo de *back propagation* [4].

## 2.8 *Boosted decision trees (BDTs)*

Se trata de uma árvore binária na qual as decisões são tomadas para uma variável por vez, até que o critério de parada seja satisfeito. Os vários nós de *output* da árvore são chamados de folhas (*leafs*). Uma representação esquemática de um árvore de decisão se encontra na figura 3 [4].

## 2.9 Definição de métricas e notações

Algumas definições de métricas (utilizadas para avaliar o desempenho) e notações importantes feitas no *paper* [4] são:

-Viés fotométrico: $\delta_{gal} = z_{phot} - z_{spec}$ [4];

-Espalhamento fotométrico: desvio padrão de $\delta_{gal}$ para um conjunto de galáxias [4];

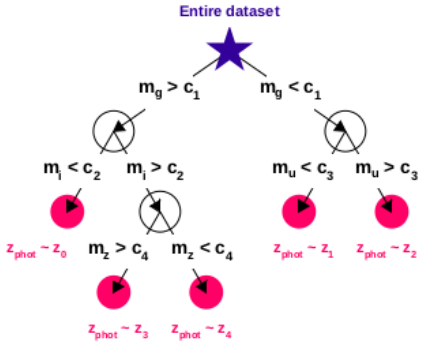


Figura 3: Representação esquemática de uma árvore de decisão. Nela, o nó raiz inicial está marcado com uma estrela, os nós internos com círculos vazios e os nós de *output* (*leafs*) com círculos preenchidos. Uma sequência de decisões binárias usando magnitudes $m_u$, $m_g$, $m_i$ e $m_z$ como variáveis de *input* é aplicada a cada elemento do conjunto de treinamento. Cada decisão binária utiliza a variável que, naquele nó, representa o melhor resultado. Referência: "*ANNz2 - Photometric Redshift and Probability Distribution Function Estimation Using Machine Learning*"[4].

-$\sigma_{68}$ denota a meia largura da área que abrange o pico do 68º percentil da distribuição de $\delta_{gal}$ [4];

-Fração atípica da distribuição do viés: $f(\alpha\sigma)$, definida como a porcentagem de objetos que têm viés maior que um fator, $\alpha$, de $\sigma$ ou $\sigma_{68}$ [4];

-Fração atípica combinada: $f(2, 3\sigma_{68} = \frac{1}{2}(f(2\sigma_{68}) + f(3\sigma_{68})$ [4].

## 2.10 Modos de treinamento do ANNz2

O ANNz2 utiliza tanto técnicas de regressão quanto técnicas de classificação. Os modos de treinamento do ANNz2 são *Single Regression*, *Random Regression*, *Binned Classification*, *Single Classification* e *Random Classification*. Esse modos de treinamento são descritos nas subseções a seguir.

### 2.10.1 Single Regression

Essa é a configuração mais simples do ANNz2. Nesse caso, uma única regressão é feita [10]. O método e a configuração do método é fixa e, portanto, não otimizada.

### 2.10.2 Random Regression

Na Regressão Aleatória (*Random Regression*), um conjunto de métodos de regressão é automaticamente gerado, sendo que esse métodos de *Machine Learning* diferem uns dos outros de diversas maneiras, como por exemplo no conjunto de parâmetros de entrada usado no treinamento. Assim que o treinamento ocorre, a otimização é realizada, obtendo-se uma distribuição de soluções do photo-z para cada galáxia e, então, essas soluções são analizadas para se determinar quais foram os métodos que atingiram performance ótima. Esses métodos selecionados então são acrescidos de suas respectivas incertezas e um conjunto de PDFs é gerado, cada uma sendo contruída por um conjunto diferente de pesos relativos associados às componentes dos métodos. Por fim, torna-se possível selecionar a melhor solução de todos os métodos randomizados. [10]

### 2.10.3 Binned Classification

Na *Binned Classification*, o conjunto de dados de *input* é subdividido em diversos pequenos grupos (*bins* de classificação). A amostra de sinal (*signal sample*) é definida como a coleção de galáxias para as quais o *redshift* espectroscópico está dentro do intervalo do *bin*. Já a amostra de fundo (*background sample*) contém todas as galáxias cujo *redshift* espectroscópico está fora do intervalo do *bin*. Então, o algoritmo realiza o treinamento em cada *redshift bin* com um método de *machine learning* de classificação diferente e o *output* para cada *bin* é traduzido como a probabilidade de uma galáxia ter um *redshift* que se encontra no intervalo desse *bin*. A distribuição das probabilidades é, então, normalizada e se torna a PDF do photo-z para a galáxia. Por fim, é gerada uma PDF total e a média ponderada da PDF, juntamente das incertezas estimadas [10].

### 2.10.4 Single Classification

Essa é a configuração mais simples do ANNz2. Nesse caso, um classificação simples é realizada [10].

### 2.10.5 Random Classification

Assim como na *Random Regression*, um conjunto de métodos de *Machine learning* é randomizado durante o treinamento. Já na fase de otimização, ocorre o cálculo do parâmetro de separação entre as distribuições de sinal/fundo (*signal/background*) e é feito um *ranking* de todas as soluções. Então, todas ou nenhuma das soluções podem ser incluída no produto final. Adicionalmente, as incertezas de classificação são calculadas [10].

# 3 Astropy: utilizando a base de dados do SDSS

A base de dados utilizada foi a do *Sloan Digital Sky Survey* (SDSS), especificamente o *release* 16 (DR16). A amostra que foi utilizada é a amostra preparada para as análises de LSS do eBOSS e inclui apenas galáxias luminosas vermelhas, tendo sido aplicados cortes e máscaras [14]. Primeiramente, foi necessário fazer o tratamento desses dados usando o Astropy para fazer um teste com o código ANNz2.

Em primeiro lugar, foi necessário fazer o *download* dos seguintes arquivos *fits* do catálogo *:

eBOSS_LRGpCMASS_clustering_data-NGC-vDR16.fits (dados Norte),
eBOSS_LRGpCMASS_clustering_data-SGC-vDR16.fits (dados Sul) e
eBOSS_LRG_full_ALLdata-vDR16.fits (dados completos).

Os dados Norte (NGC) e o dados Sul (SGC) possuem uma coluna chamada 'LRG_ID' que serve para identificar os objetos presentes no catálogo. Então, após definir 'catalog_dir' como o PATH do local em que se encontram os arquivos citados, foram executadas as seguintes linhas de código para selecionar apenas as linhas com 'LRG_ID' válidos.

```
NGC=os.path.join(catalog_dir, "eBOSS_LRGpCMASS_clustering_data-NGC-vDR16.fits")
SGC=os.path.join(catalog_dir, "eBOSS_LRGpCMASS_clustering_data-SGC-vDR16.fits")
ALL=os.path.join(catalog_dir, "eBOSS_LRG_full_ALLdata-vDR16.fits")

evt1_data = Table.read(NGC, memmap=True)
```

*Disponível em: https://data.sdss.org/sas/dr17/eboss/lss/catalogs/DR16/

```
evt2_data = Table.read(SGC, memmap=True)
all_data = Table.read(ALL, memmap=True)

#verificando o tamanho da amostra:
(~evt1_data['LRG_ID'].data.mask).sum(),
(~evt2_data['LRG_ID'].data.mask).sum(), len(all_data)

mask_evt1_lrg_only = ~evt1_data['LRG_ID'].data.mask
mask_evt2_lrg_only = ~evt2_data['LRG_ID'].data.mask
```

O próximo passo foi fazer o *join*, individualmente, dos arquivos NGC e SGC com o arquivo ALL DATA. Feito isso, foi feito o *stack* (empilhamento) dos dois conjuntos obtidos e, assim, gerados um arquivo com todos os dados NGC e SGC com 'LRG_ID' válidos e um arquivo guardando as informações de identificação dos objetos.

```
lrg1_joined = join(evt1_data[mask_evt1_lrg_only],
all_data[['RUN','CAMCOL','FIELD','ID','RERUN','LRG_ID','MODELMAG']], keys='LRG_ID')

lrg2_joined = join(evt2_data[mask_evt2_lrg_only],
all_data[['RUN','CAMCOL','FIELD','ID','RERUN','LRG_ID','MODELMAG']], keys='LRG_ID')

lrg_joined = vstack([lrg1_joined, lrg2_joined])

lrg_joined.write(os.path.join(catalog_dir,
'eBOSS_LRG_clustering_modelmag_data-NGCSGC-vDR16.fits'), overwrite=True)

lrg_joined[['LRG_ID', 'RA', 'DEC', 'RUN','CAMCOL','FIELD','ID',
'RERUN']].write('eBOSS_LRG_clustering_-NGCSGC-vDR16_obj_ids.csv', overwrite=True)
```

Em seguida, selecionando o arquivo 'eBOSS_LRG_clustering_-NGCSGC-vDR16_obj_ids.csv' e o colocando em uma tabela SQL, foi possível descobrir que o identificador dos objetos fotométrico ('objID') pode ser obtido das colunas 'RUN', 'RERUN', 'CAMCOL', 'FIELD' e 'ID' com a função 'fObjidFromSDSS' do SkyServer.

Foi executada, então, a seguinte *query*.

```
SELECT
    c.lrg_id,
    p.objID, p.specObjID, p.ra, p.dec,
```

```
    p.dered_u, p.dered_g, p.dered_r, p.dered_i, p.dered_z,
    p.err_u, p.err_g, p.err_r, p.err_i, p.err_z
FROM MyDB.ebossDR16LRGClustering as c
   JOIN DR18.PhotoObj p ON p.objID = dbo.fObjidFromSDSS(2, c.run, c.rerun, c.camcol,
   c.field, c.id)
INTO MyDB.ebossDR16LRGClusteringPhotometry
```

Após baixar a tabela 'ebossDR16LRGClusteringPhotometry' no arquivo 'ebossDR16LRGClusteringPhotometry.csv' e esse arquivo foi colocado no diretório com os demais arquivos. Assim, foi o *join* (com relação a 'LRG_ID' desse arquivo de fotometria com o arquivo obtido anteriormente pelo empilhamento dos arquivos NGC e SGC.

```
lrg_joined = os.path.join(catalog_dir,
'eBOSS_LRG_clustering_modelmag_data-NGCSGC-vDR16.fits')

lrg_photo = os.path.join(catalog_dir, 'ebossDR16LRGClusteringPhotometry.csv')

lrg_joined_tab = Table.read(lrg_joined)
lrg_photo_tab = Table.read(lrg_photo)

lrg_joined_join = join(lrg_photo_tab, lrg_joined_tab, keys_left='lrg_id',
keys_right='LRG_ID')

np.abs(lrg_joined_join['ra'] - lrg_joined_join['RA']).max()
np.abs(lrg_joined_join['dec'] - lrg_joined_join['DEC']).max()
```

A tabela 'lrg_joined_join' possui 174816 linhas e 36 colunas. No entanto, as únicas colunas que são necessárias são: 'MODELMAG', 'err_u', 'err_g', 'err_r', 'err_i', 'err_z' e 'Z'. Assim, foi necessário criar um novo arquivo somente com os dados necessários para o *input* do ANNz2.

```
new_order = ['MODELMAG', 'lrg_id','err_u','err_g','err_r','err_i','err_z','Z']

lrg_joined_join_selected = lrg_joined_join[new_order]

lrg_joined_join_selected.write(os.path.join(catalog_dir,
'eBOSS_LRG_clustering_modelmag_data-NGCSGC-selected-vDR16.fits'), "overwrite=True")
```

A tabela 'lrg_joined_join_selected' está representada na Figura 4.

Table length=174816

| MODELMAG | lrg_id | err_u | err_g | err_r | err_i | err_z | Z |
|---|---|---|---|---|---|---|---|
| float32[5] | int64 | float64 | float64 | float64 | float64 | float64 | float64 |
| 25.723623 .. 19.939554 | 155 | 1.166705 | 0.1577692 | 0.1553537 | 0.09805954 | 0.1606366 | 0.6899790890408843 |
| 24.98986 .. 19.97161 | 164 | 1.683199 | 0.475468 | 0.1872905 | 0.0872043 | 0.1701971 | 0.6703961690458896 |
| 21.900898 .. 19.842785 | 166 | 0.4250453 | 0.3084107 | 0.2416351 | 0.1275055 | 0.2263277 | 0.938681613552569 |
| 24.926273 .. 19.667055 | 167 | 2.522411 | 0.2480161 | 0.295671 | 0.1257648 | 0.1927489 | 0.8546939792986834 |
| 25.94632 .. 19.737507 | 178 | 1.070446 | 0.6998233 | 0.1665629 | 0.07360715 | 0.152274 | 0.7486357721889371 |
| 24.903866 .. 19.478413 | 182 | 2.791524 | 0.380014 | 0.14156 | 0.08432812 | 0.1603149 | 0.6099426425190794 |
| 25.382797 .. 19.59364 | 185 | 2.208333 | 0.3695331 | 0.3637002 | 0.1716594 | 0.1684795 | 0.8808363995653804 |
| 22.071062 .. 19.809128 | 188 | 0.378562 | 0.2430625 | 0.1393567 | 0.08222785 | 0.1445032 | 0.6468334192655131 |
| 23.214466 .. 19.976397 | 189 | 0.9849465 | 0.5371121 | 0.1710789 | 0.09241519 | 0.1620435 | 0.8383725926612978 |
| 26.460602 .. 19.471415 | 588 | 0.9072875 | 0.3969019 | 0.1793048 | 0.1076104 | 0.1508087 | 0.6412123656670905 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 26.019072 .. 19.18466 | 377612 | 1.411682 | 0.3379578 | 0.165506 | 0.1082411 | 0.1857756 | 0.6861845038655601 |
| 26.408495 .. 19.759415 | 377613 | 0.5933867 | 0.4745699 | 0.1337036 | 0.06912997 | 0.1758418 | 0.7253840315689559 |
| 24.17064 .. 19.419062 | 377614 | 1.211575 | 0.3121546 | 0.08819267 | 0.04476507 | 0.09807403 | 0.6276900965299482 |
| 24.974031 .. 19.884604 | 377615 | 1.233365 | 0.2250503 | 0.1139753 | 0.06401406 | 0.1407673 | 0.6245972507066833 |
| 25.447977 .. 19.748634 | 377616 | 0.9833993 | 0.7299526 | 0.1167653 | 0.05342529 | 0.1201108 | 0.7136041503853774 |
| 26.533506 .. 19.559748 | 377620 | 0.4251032 | 0.2760071 | 0.09417978 | 0.04830667 | 0.1177817 | 0.6293471585204776 |
| 21.407385 .. 19.715363 | 377621 | 0.2416606 | 1.129584 | 0.1895795 | 0.1073512 | 0.2094158 | 0.8315863916537576 |
| 23.424105 .. 19.952333 | 377625 | 0.7521243 | 0.6160727 | 0.1903158 | 0.07554701 | 0.1395938 | 0.7494615135042955 |
| 22.829556 .. 19.956715 | 377628 | 0.7297657 | 0.605174 | 0.3218153 | 0.1291861 | 0.2375163 | 0.8280102785040316 |
| 26.677126 .. 19.941578 | 377631 | 0.4600784 | 0.5767424 | 0.201619 | 0.1212205 | 0.1656556 | 0.6895709834159385 |

Figura 4: Tabela 'lrg_joined_join_selected'

Usando 'novo' para abrir o arquivo 'eBOSS_LRG_clustering _modelmag_data-NGCSGC-selected-vDR16.fits', foi necessário, então, separar as magnitudes 'U', 'G', 'R', 'I', 'Z' de 'MODELMAG'.

```
novo = fits.open("/home/sofia/Documentos/Astropy/
eBOSS_LRG_clustering_modelmag_data-NGCSGC-selected-vDR16.fits")

data1 = novo[1].data
MODELMAG = data1['MODELMAG']
LRG_ID = data1['LRG_ID']

M0 = [MODELMAG[n][0] for n in range(len(MODELMAG))]
lrg_joined_join_selected['#F:MAG_U'] = M0
lrg_joined_join_selected['F:MAGERR_U'] = lrg_joined_join_selected['err_u']

M1 = [MODELMAG[n][1] for n in range(len(MODELMAG))]
lrg_joined_join_selected['F:MAG_G'] = M1
lrg_joined_join_selected['F:MAGERR_G'] = lrg_joined_join_selected['err_g']

M2 = [MODELMAG[n][2] for n in range(len(MODELMAG))]
lrg_joined_join_selected['F:MAG_R'] = M2
lrg_joined_join_selected['F:MAGERR_R'] = lrg_joined_join_selected['err_r']
```

```
M3 = [MODELMAG[n][3] for n in range(len(MODELMAG))]
lrg_joined_join_selected['F:MAG_I'] = M3
lrg_joined_join_selected['F:MAGERR_I'] = lrg_joined_join_selected['err_i']

M4 = [MODELMAG[n][4] for n in range(len(MODELMAG))]
lrg_joined_join_selected['F:MAG_Z'] = M4
lrg_joined_join_selected['F:MAGERR_Z'] = lrg_joined_join_selected['err_z']

del lrg_joined_join_selected['MODELMAG']
del lrg_joined_join_selected['err_u']
del lrg_joined_join_selected['err_g']
del lrg_joined_join_selected['err_r']
del lrg_joined_join_selected['err_i']
del lrg_joined_join_selected['err_z']
del lrg_joined_join_selected['lrg_id']
lrg_joined_join_selected['zz'] = lrg_joined_join_selected['Z']
del lrg_joined_join_selected['Z']
lrg_joined_join_selected.rename_column('zz', 'D:Z')

lrg_joined_join_selected.write(os.path.join(catalog_dir,
'eBOSS_LRG_clustering_modelmag_data-NGCSGC-selected-and-organized-vDR16.csv'))
```

Assim, foi gerado também, no formato de entrada do ANNz2 (csv), o arquivo contendo todas as informações n ecessárias para realizar *Single Regression* e *Random Regression*.

Por fim, foi necessário dividir o conjunto de dados de 174816 objetos em vários subconjuntos para avaliação, treinamento e otimização.

# 4 Resultados

## 4.1 Single Regression

Após a execução do modo Single Regression com os arquivos adequados, com o intervalo de $0.6 < z < 1.0$ operando com o modo ANN, foi possível obter uma tabela com as colunas 'D:Z' (*redshift* verdadeiro, no caso o espectroscópioc) e 'F:ANNZ_best'

(melhor estimativa de *redshift* obtida após, treinamento, otimização e validação).

Plotando o gráfico de $z$ (que é o *redshift* espectroscópico, $z_{esp}$) por $z_{best}$ (que é o *redshift* fotométrico, $z_{photo}$), obtemos a Figura 5.

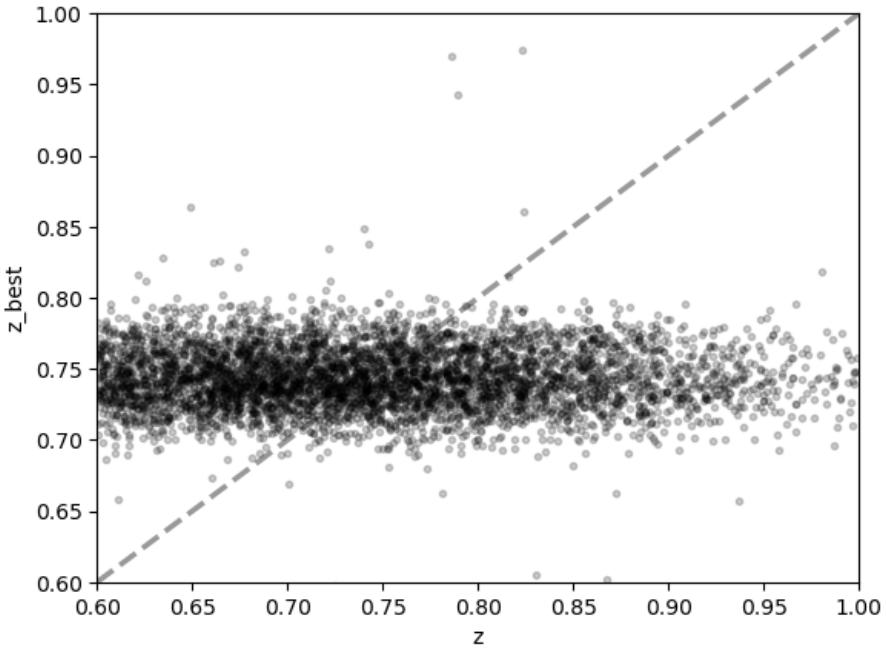


Figura 5: Gráfico de $z$ versus $z_{best}$ para Single Regression.

Claramente, o treinamento não foi muito bem sucedido. O treinamento perfeito ocorreria caso o gráfico se aproximasse da equação $z = z_{best}$. No caso, os valores de $z_{best}$ estão densamente concentrados entre 0.7 e 0.8, apontando para a aproximação linear de uma reta constante em torno de 0.75.

Também plotando os gráficos de $\delta$, $\sigma$ e $f_{2\sigma}$, obteve-se a Figura 6.

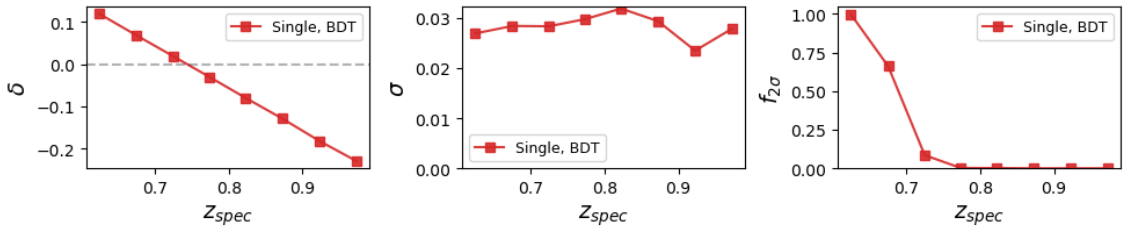


Figura 6: Gráficos de $\delta$, $\sigma$ e $f_{2\sigma}$ após Single Regression.

No entanto, os gráficos obtido com o *input* do '*examples*' do próprio código original, foram os que podem ser visualizados nas Figuras 7 e 8. É possível observar que os resultados foram parecidos com os obtidos utilizando o DR16 do SDSS no modo Single

Regression no modo ANN, no entanto bem mais dispersos, estando dentro do intervalo $0.50 < z_{best} < 0.75$.

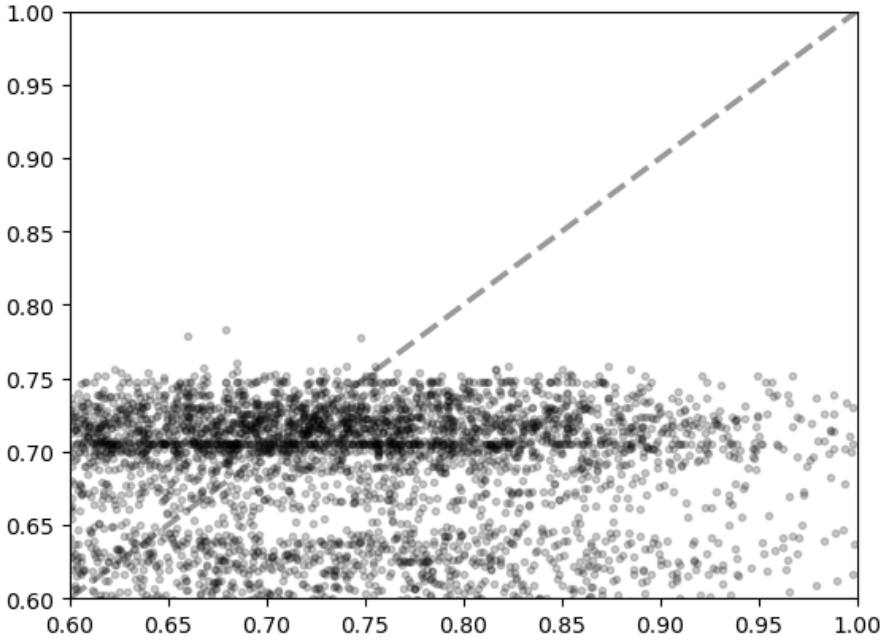


Figura 7: Gráfico de $z$ versus $z_{best}$ para Single Regression utilizando os dados do '*examples*'.

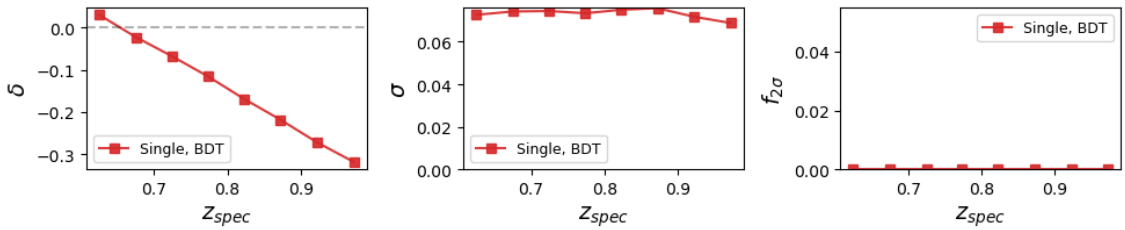


Figura 8: Gráficos de $\delta$, $\sigma$ e $f_{2\sigma}$ após Single Regression utilizando os dados do '*examples*'.

## 4.2 Random Regression

Após a execução do modo Random Regression com os arquivos adequados, com o intervalo de $0.6 < z < 1.0$ operando com o modo BDT, com 50 MLMs, foi possível obter uma tabela com as colunas 'D:Z' (*redshift* verdadeiro, no caso o espectroscópico) e 'F:ANNZ_best' (melhor estimativa de *redshift* obtida após, treinamento, otimização e validação).

Plotando o gráfico de $z$ por $z_{best}$, obtemos a Figura 9.

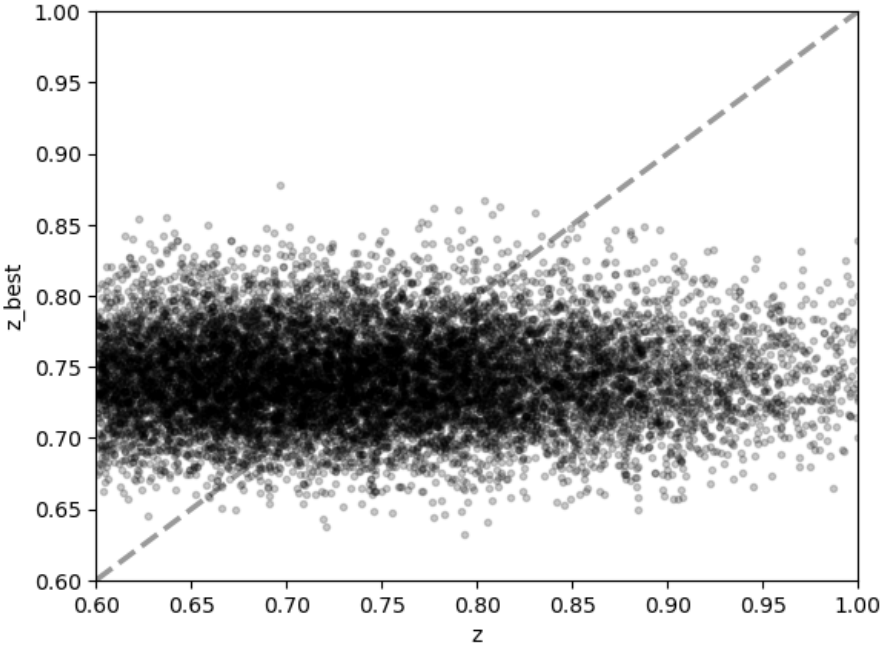


Figura 9: Gráfico de $z$ versus $z_{best}$ para Random Regression.

No entanto, o gráfico obtido com o *input* do '*examples*' do próprio código original, foi o que pode ser visualizado na Figura 10. É possível observar que os resultados foram bastante diferentes dos obtidos utilizando o DR16 do SDSS no modo Random Regression no modo BDT. Existem algumas hipóteses sobre o porque disso ter acontecido. Talvez a base de dados selecionada do DR16 do SDSS tenha sido pouco representativa (os resultados estão todos em torno do intervalo de *redshift* $0.65 < z < 0.80$, de forma que pode ser que a quantidade de objetos nessa faixa de *redshift* nas amostras utilizadas pode ser muito mais alta do que a quantidade de outras faixas de *redshift*) ou talvez tenha acontecido *overtraining*. Será necessária uma investigação mais cuidadosa para descobrir o que pode estar causando tamanha diferença.

# 5 Conclusão

Com tudo que foi discutido até aqui, fica nítido que será necessário, além de refazer testes com *Single Regression* e *Random Regression* e investigar o motivo de o treinamento não estar ocorrendo como esperado, testar os demais modos de operação *Random Classification*, *Binned Classification* e *Single Classification*) para realizar com-

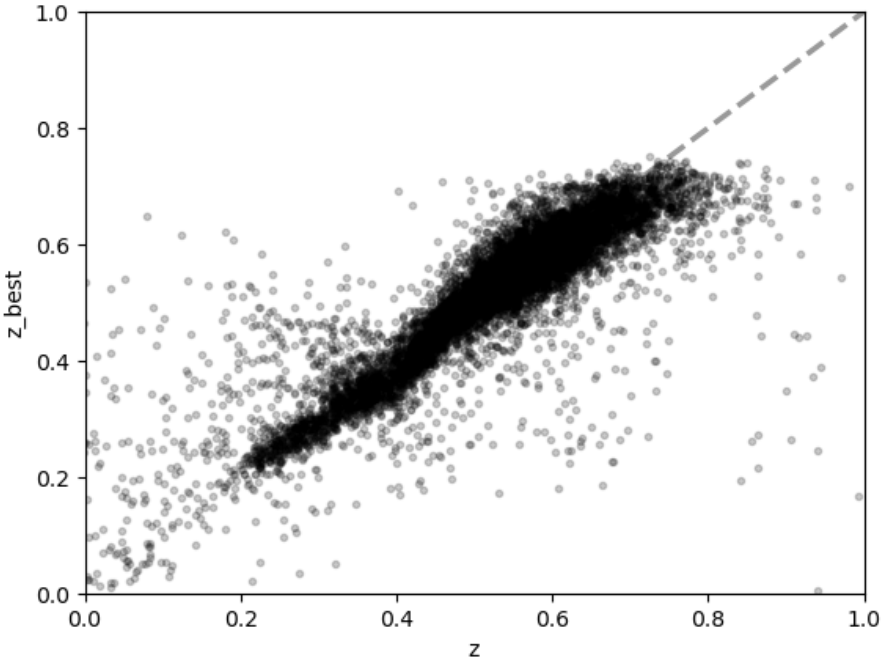


Figura 10: Gráfico de $z$ versus $z_{best}$ para Random Regression utilizando o 'examples' original.

parações entre os resultados obtidos com a *Single regression* e com a *Random Regression* e possivelmente obter evidência de um treinamento bastante bem sucedido, com um gráfico $z$ versus $z_{best}$ mais próximo da reta $z = z_{best}$.

# Referências


[1] OLIVEIRA, Kepler de. Astronomia e Astrofísica. 3ª ed. São Paulo: Editora Livraria da Física, 2014;

[2] NEWMANN, Jeffrey A., GRUEN, Daniel. Photometric Redshifts for Next-Generation Surveys. June, 2022. Disponível em: https://arxiv.org/pdf/2206.13633.pdf

[3] COLLISTER, Adrian A., LAHAV, Ofer. ANNz: Estimating Photometric Redshifts Using Artificial Neural Networks. University of Cambridge, Cambridge, UK, p. (1,6), February, 2004. Disponível em: https://arxiv.org/pdf/astro-ph/0311058.pdf

[4] SADEH, I., ABDALLA, F. B., LAHAV, O. ANNz2 - Photometric Redshift and Probability Distribution Function Estimation Using Machine Learning. University College London, UK; Rhodes University, PO, p. (1,22), June, 2016. Disponível em: https://arxiv.org/abs/1507.00490

[5] D.G. York et al. The Sloan Digital Sky Survey: Tecnical Summary. AJ, 120:1579, 2000.

[6] B. Abareshi et al. Overview of the Instrumentation for the Dark Energy Spectroscopic Instrument. AJ, 164:207.

[7] T Abbolt et al. The Dark Energy Survey: more than dark energy - an overview. MRAS, 460 1270 D, 2016.

[8] Z. Ivezic et al. LSST: From Science Drivers to Reference Design and Anticipated Data Product. ApJ, 873 111, 2019.

[9] Sistemas de Coordenadas - UFRGS, Disponível em: http://astro.if.ufrgs.br/coord.htm

[10] Código e documentação do ANNz2 disponível para instalação. Disponível em: https://github.com/IftachSadeh/ANNZ

[11] IVEZIC, Z., CONNOLLY, A., VANDERPLAS, J., GRAY, A. Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data. Princeton University Press, 2014.

[12] ROOT - *Data Analysis Framework*, 2024. Disponível em: [https://root.cern/manual/tmva/]. Acesso em: 9, março, 2024.

[13] GÉRON, Aurélien. Hands-On Machine Learning with Scikit-Learn and TensorFlow: concepts, tools and techniques to build intelligent systems. O'Reilly Media, 2017.

[14] ROSS, Aschley J., etc, The Completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: Large-scale structure catalogues for cosmological analysis.https://arxiv.org/abs/2007.09000 September, 2020, Disponível em: https://arxiv.org/abs/2007.09000.