Avaliação de Modelos de Regressão e Redes Neurais na Estimativa de Redshift: Limitações e Potencial

1st Carlos Vinícius S. Mesquita
dept. de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brazil
carlos.mesquita12@alu.ufc.br

2nd Francisco Vinicius Castro Silveira
dept. de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Ceará
viniciuscastro2615@alu.ufc.br

***Abstract*—Este trabalho investiga a capacidade de diferentes modelos de regressão em estimar uma variável astronômica contínua a partir de medidas fotométricas com o uso de dados do Sloan Digital Sky Survey (SDSS DR17). Foram implementados modelos de Regressão Linear Ordinária (OLS), Regressão de Ridge, Regressão por Componentes Principais (PCR) e uma Rede Neural Multicamadas, utilizando procedimentos de pré-processamento e validação cruzada. Os resultados mostram que, embora os modelos capturem parte da variabilidade dos dados, o poder preditivo permanece limitado, sugerindo que a relação entre os preditores disponíveis e a variável resposta é fraca ou pouco linear. Ainda assim, o estudo evidencia diferenças de desempenho entre os métodos e discute a adequação de cada abordagem ao problema.**

***Index Terms*—Redes Neurais, Redshift, Regressão.**

I. Introducão

Galáxias, estrelas e quasares são objetos centrais para a astronomia moderna, pois representam diferentes estágios de evolução cósmica e permitem investigar fenômenos fundamentais do Universo. Esses sistemas exibem propriedades físicas diversas — como brilho, composição química e distribuição espacial — que podem ser medidas por meio de técnicas observacionais e registradas em grandes catálogos astronômicos. Tais catálogos reúnem dados fotométricos e espectrais provenientes de observações sistemáticas, possibilitando a aplicação de métodos estatísticos avançados na análise desses objetos.

Entre as grandezas observacionais mais relevantes, o redshift (desvio para o vermelho) ocupa papel de destaque, por representar um indicador direto da distância cosmológica e, portanto, da própria expansão do Universo. Em muitos levantamentos astronômicos, esse valor pode ser estimado a partir de medidas fotométricas multibanda, embora a relação entre essas variáveis frequentemente seja não linear e influenciada por ruídos e incertezas inerentes ao processo de medição.

Nesse contexto, técnicas de regressão estatística e de aprendizado de máquina tornam-se ferramentas essenciais para investigar até que ponto as características observadas dos objetos celestes permitem prever variáveis físicas de interesse. Modelos como Regressão Linear Ordinária (OLS), Regressão de Ridge, Regressão por Componentes Principais (PCR) e redes neurais multicamadas (MLP) oferecem abordagens complementares para explorar tais relações, permitindo comparar diferentes formas de modelar dependências lineares, regularizar parâmetros e capturar possíveis padrões não lineares.

Assim, este trabalho examina a capacidade preditiva desses modelos ao estimar uma variável astronômica contínua a partir de atributos fotométricos e estruturais presentes em um conjunto de dados real. Além de avaliar o desempenho final dos métodos, o estudo discute suas limitações, a influência da dimensionalidade e os impactos de técnicas como padronização e redução de dimensionalidade. Também é avaliada a capacidade dos modelos de generalizar para novos dados, buscando identificar configurações que equilibram desempenho e baixo overfitting — isto é, modelos que não apenas se ajustam bem ao conjunto de treino, mas mantêm coerência e estabilidade ao prever dados inéditos. O objetivo é compreender a viabilidade de diferentes abordagens de regressão nesse tipo de cenário e avaliar em que medida o conjunto de dados disponível sustenta predições robustas e generalizáveis.

Para avaliar o desempenho dos modelos regressivos desenvolvidos neste trabalho, utilizamos duas métricas amplamente empregadas em problemas de predição contínua: o Root Mean Squared Error (RMSE) e o coeficiente de determinação (R^2). O RMSE quantifica o erro médio das previsões, considerando a magnitude dos resíduos. Ele mede, em média, o quanto as previsões $\hat{y}$se desviam dos valores reais y. Sua definição é dada por:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Valores menores de RMSE indicam previsões mais precisas, sendo uma métrica sensível a erros maiores devido ao termo quadrático. Já o coeficiente de determinação R^2 expressa a proporção da variabilidade do alvo que é explicada pelo modelo. Essa métrica compara o desempenho do modelo com o de uma predição ingênua baseada na média dos valores

observados. Sua formulação é:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Valores de R^2 próximos de 1 indicam alta capacidade explicativa, enquanto valores próximos (ou menores) de 0 sugerem que o modelo não captura adequadamente a estrutura dos dados.

O uso conjunto dessas duas métricas permite uma avaliação complementar: enquanto o RMSE quantifica o erro em escala absoluta, o R^2 revela a qualidade relativa do ajuste do modelo em relação à variabilidade dos dados.

II. Metodos

Neste trabalho, utilizamos um conjunto de preditores transformados a partir do conjunto de treino para a construção e comparação de diferentes modelos de regressão. Inicialmente, foi realizada uma análise exploratória concisa e um pré-processamento básico, aplicado igualmente ao treino e ao teste, envolvendo remoção de colunas irrelevantes e transformação dos preditores para reduzir assimetrias e padronizar escalas. Em seguida, desenvolvemos e avaliamos quatro classes de modelos regressivos: (i) Regressão Linear via OLS, implementada tanto manualmente quanto por funções prontas, incluindo uma validação cruzada k-fold; (ii) Regressão Ridge, também implementada do zero e com biblioteca, na qual o parâmetro de regularização λ foi escolhido por validação cruzada; (iii) PCR (Principal Component Regression), utilizando PCA para reduzir dimensionalidade e seleção do número ótimo de componentes por validação cruzada; e (iv) uma Rede Neural Multicamadas (MLP), treinada sobre os mesmos preditores transformados para investigar possíveis relações não lineares entre as variáveis. Todas as etapas foram conduzidas aplicando sempre o mesmo pré-processamento do conjunto de treino ao conjunto de teste, garantindo consistência metodológica entre os modelos comparados.

O trabalho foi desenvolvido em Python, escolhido devido ao seu amplo ecossistema voltado a análise exploratória de dados (EDA) e à disponibilidade de ferramentas consolidadas para modelagem estatística e aprendizado de máquina. Para manipulação e limpeza do dataset utilizamos a biblioteca Pandas, enquanto NumPy foi empregada para operações matriciais e cálculos numéricos de baixa complexidade. A etapa de pré-processamento, incluindo padronização e transformações nos preditores, fez uso de módulos do scikit-learn, que também foi utilizado para as implementações de referência dos modelos OLS, Ridge, PCR e MLP. Para a construção manual de alguns modelos e funções auxiliares, utilizamos operações vetorizadas de NumPy. A visualização das curvas de aprendizado, métricas e gráficos de desempenho foi realizada por meio da biblioteca Matplotlib permitindo uma análise clara e comparativa dos modelos obtidos.

A. Descrição do Dataset

O conjunto de dados original contém 100 000 objetos astronômicos descritos por 18 variáveis. Esses atributos incluem identificadores, posições celestes, magnitudes em diferentes bandas e informações do processo de observação. A Tabela I apresenta um resumo das colunas fornecidas no dataset.

TABLE I
DESCRIÇÃO DAS COLUNAS DO CONJUNTO DE DADOS SDSS DR17.

Coluna	Descrição
`obj_ID`	Identificador único do objeto no catálogo de imagens.
`alpha`	Ascensão Reta (Right Ascension) em coordenadas equatoriais.
`delta`	Declinação (Declination) em coordenadas equatoriais.
`u`	Magnitude no filtro ultravioleta (u).
`g`	Magnitude no filtro verde (g).
`r`	Magnitude no filtro vermelho (r).
`i`	Magnitude no filtro infravermelho próximo (i).
`z`	Magnitude no filtro infravermelho (z).
`run_ID`	Número da varredura do telescópio.
`rerun_ID`	Identifica o processamento da imagem.
`cam_col`	Coluna da câmera (CCD).
`field_ID`	Campo observado.
`spec_obj_ID`	Identificador único do objeto no catálogo espectroscópico.
`class`	Classe: galáxia, estrela ou quasar.
`redshift`	Desvio para o vermelho medido.
`plate`	Identificador da placa espectroscópica.
`MJD`	Data Juliana Modificada da captura.
`fiber_ID`	Fibra óptica usada na captura.

Para este trabalho, utilizamos apenas as variáveis numéricas relacionadas à posição, magnitudes e redshift. Os identificadores foram descartados por não contribuírem para análise estatística, e as colunas `class`, `plate` e `MJD` foram removidas por não apresentarem relação com a variável-alvo. Uma observação foi eliminada por conter valores anômalos.

As colunas `class`, `plate` e `MJD` também foram removidas antes do treinamento dos modelos. A variável `class` é categórica (STAR, GALAXY, QSO) e não contribui para a tarefa de regressão voltada à estimativa do redshift. Já `plate` representa apenas um identificador do espectrógrafo, sem relação física com a variável-alvo. Por fim, `MJD` corresponde ao instante da observação e não apresenta impacto significativo na previsão do redshift, podendo inclusive introduzir ruído adicional ao modelo.

Uma observação foi removida por apresentar valores discrepantes na inspeção exploratória, o conjunto final utilizado contém $N = 99\,999$ registros, $D = 10$ variáveis explicativas e $L = 3$ classes distintas presentes no dataset original (galáxias, estrelas e quasares).

B. Regressão linear via OLS (Ordinary Least Squares)

A Regressão Linear OLS foi utilizada como modelo base para estabelecer uma referência de desempenho para os métodos subsequentes. O OLS estima os coeficientes β que minimizam a soma dos erros quadráticos entre as previsões e os valores reais. Sua solução fechada é dada por:

$$\hat{\beta} = (X^\top X)^{-1} X^\top y$$

onde X é a matriz de preditores (com intercepto incluído) e y é o vetor da variável resposta (redshift). Implementamos o OLS de duas formas:

Na implementação do zero, utilizamos operações matriciais com NumPy para construir toda a solução do OLS manualmente. Inicialmente, foi adicionada uma coluna de intercepto

à matriz de preditores e, em seguida, calculada a matriz $X^T X$, cuja inversa foi utilizada para obter os coeficientes $\hat{\beta}$ de acordo com a solução fechada do modelo. Com os coeficientes estimados, geramos previsões tanto no conjunto de treino quanto no conjunto de teste e, posteriormente, calculamos manualmente as métricas de desempenho $RMSE$ e R^2, Paralelamente, implementamos também a versão com biblioteca, utilizando a classe $LinearRegression$ do scikit-learn como referência.

Para além da divisão treino-teste (80/20), aplicamos também uma validação cruzada 10-fold para estimar a capacidade de generalização do modelo e reduzir a dependência de uma única partição dos dados. Em cada um dos 10 folds, o modelo foi treinado em 9 partições e avaliado na partição restante,Adotamos 10-fold cross-validation em vez de 5-fold porque o conjunto de treino é grande (80.000 observações), permitindo dividir em mais folds sem prejuízo estatístico. Além disso, o 10-fold tende a produzir estimativas de erro mais estáveis e com menor viés, já que o modelo é treinado, a cada iteração, com cerca de 90% dos dados. A validação foi aplicada tanto na implementação manual quanto na que houve o uso de biblioteca, e os valores médios de RMSE e R² foram utilizados como estimativas mais robustas do desempenho.

C. Regressão de Ridge

A Regressão Ridge foi implementada inicialmente do zero utilizando apenas operações matriciais com NumPy. Primeiro, foi adicionada manualmente uma coluna de intercepto à matriz de preditores. Em seguida, o vetor de coeficientes foi calculado pela solução fechada:

$$\hat{\beta} = (X^\top X + \lambda I)^{-1} X^\top y$$

com a modificação de que o termo de intercepto não foi penalizado. Após obter os coeficientes, foram geradas as previsões para treino e teste, e as métricas RMSE e R^2 foram calculadas manualmente para avaliação. Em seguida houve a implementação com a biblioteca scikit-learn utilizando a classe Ridge

O valor ótimo de λ (parâmetro de penalização da Ridge) foi selecionado utilizando 10-fold cross-validation, tanto na implementação do zero quanto na versão com scikit-learn. Em ambos os casos, diferentes valores de λ foram testados e aquele que minimizou o RMSE médio entre os folds foi escolhido. Esse procedimento foi adotado porque o parâmetro λ controla diretamente o equilíbrio entre viés e variância do modelo. Assim, a cross-validation fornece uma estimativa mais robusta do erro de generalização, reduzindo o risco de overfitting e garantindo que o valor escolhido funcione bem não apenas nos dados de treino, mas também em novos dados.

D. PCR (Principal Component Regression)

Após o pré-processamento, aplicou-se PCA aos preditores padronizados. A transformação fundamental utilizada é dada por

$$Z = TV^\top,$$

onde Z é a matriz de preditores padronizada, V contém os autovetores (direções principais) e T representa os dados projetados nessas novas direções. Essa decomposição permite reescrever os preditores em termos de componentes ortogonais, reduzindo colinearidade e concentrando a variância relevante em poucas dimensões.

Para escolher quantos componentes utilizar no modelo de regressão, empregou-se validação cruzada 10-fold, avaliando o RMSE médio para diferentes valores de k. O número ótimo foi definido como aquele que minimizou o erro de validação. Com esse valor, ajustou-se o modelo PCR final no conjunto de treinamento e projetou-se o conjunto de teste na mesma base, permitindo avaliar o desempenho final por meio das métricas RMSE e R^2.

O número ótimo de componentes principais foi selecionado por meio de validação cruzada 10-fold, sendo definido como aquele que minimizou o RMSE médio. Esse procedimento garante que o modelo final minimize o erro de predição esperado, evitando tanto a alta variância associada ao uso de muitos componentes quanto o aumento de viés decorrente de poucos componentes.

Com o número ótimo k^* determinado, ajustou-se o modelo PCR ao conjunto de treinamento projetado, e o conjunto de teste foi transformado usando a mesma PCA, garantindo consistência entre as etapas. Os valores de RMSE e R^2 obtidos para treino e teste mostraram-se próximos, indicando bom equilíbrio entre viés e variância e ausência de sobreajuste significativo.

Com o número ótimo k^* determinado, ajustou-se o modelo PCR ao conjunto de treinamento projetado e o conjunto de teste foi transformado usando a mesma PCA, garantindo consistência entre as etapas.

A aplicação do PCR mostrou-se adequada para o conjunto de dados analisado, especialmente por lidar bem com a forte colinearidade entre os preditores originais. O uso do PCA permitiu reduzir o número de variáveis, mantendo a maior parte da informação importante e diminuindo redundâncias. A validação cruzada indicou que apenas uma parte dos componentes era suficiente para obter um bom desempenho, evitando modelos excessivamente complexos.

Os valores de RMSE e R^2 obtidos no teste foram próximos aos obtidos no treinamento, indicando que o modelo generalizou bem e não apresentou sinais relevantes de sobreajuste. Isso mostra que a combinação entre PCA e regressão linear conseguiu representar de forma eficiente a relação entre os preditores e a variável resposta.

Assim, o PCR se mostrou uma técnica confiável e eficiente para este problema, garantindo um bom equilíbrio entre simplicidade, estabilidade e desempenho preditivo.

E. Rede Neural MLP (Multilayer Perceptron)

A etapa final consistiu no desenvolvimento de um modelo de rede neural MLP para regressão, com o objetivo de investigar possíveis relações não lineares entre os preditores e o redshift. O mesmo pré-processamento aplicado aos modelos lineares foi mantido

A arquitetura adotada foi composta por três camadas escondidas com 256, 128 e 64 neurônios respectivamente, todas utilizando a função de ativação ReLU (escolhida por promover melhor estabilidade do gradiente e maior capacidade de modelagem não linear), a dimensionalidade decrescente das camadas ajuda a estabilizar o treinamento. A escolha dessa arquitetura buscou oferecer capacidade suficiente para modelar relações potencialmente complexas no conjunto de dados, sem produzir sobreajuste excessivo. A camada de saída contém um único neurônio linear, adequado a tarefas de regressão.

A predição da rede neural segue o modelo geral de um MLP, onde cada camada aplica uma transformação afim seguida da função de ativação ReLU($\mathrm{ReLU}(z) = \max(0, z)$):

$$h_1 = \mathrm{ReLU}(W_1 x + b_1),$$

$$h_2 = \mathrm{ReLU}(W_2 h_1 + b_2),$$

$$h_3 = \mathrm{ReLU}(W_3 h_2 + b_3),$$

e a saída é produzida por:

$$\hat{y} = W_4 h_3 + b_4$$

O modelo foi treinado utilizando o otimizador Adam com taxa de aprendizado inicial de $8 * 10^{-}4$, minibatch de 64 amostras e até 300 iterações. A função de erro utilizada foi o Mean Squared Error (MSE). Para evitar overfitting, ativou-se early stopping com paciência de 20 épocas, interrompendo o treinamento caso o desempenho de validação deixasse de melhorar. Após o treinamento, a rede neural foi avaliada no conjunto de teste utilizando RMSE e R^2, permitindo comparação direta com os modelos lineares implementados nas etapas anteriores.

RESULTADOS

Para organizar a apresentação dos resultados de forma mais clara e didática, a seção foi dividida em cinco partes distintas. Cada uma das quatro primeiras seções é dedicada exclusivamente a um dos modelos avaliados — Regressão Linear (OLS), Regressão Ridge, Regressão por Componentes Principais (PCR) e Rede Neural Multicamadas — permitindo analisar o desempenho individual de cada método sem sobrecarregar o leitor com comparações imediatas. Já a quinta parte reúne uma comparação geral entre todos os modelos, sintetizando suas métricas e destacando diferenças e semelhanças de desempenho.

F. OLS

Os resultados obtidos para a Regressão Linear via OLS mostraram desempenho praticamente idêntico entre a implementação feita do zero e a versão baseada na biblioteca scikit-learn. Em ambos os casos, o modelo apresentou um erro médio moderado, com RMSE de aproximadamente 0.62, demostrado na tabela II, tanto no conjunto de treino quanto no conjunto de teste. Essa proximidade sugere que não houve sobreajuste significativo e que o comportamento do modelo é consistente entre treino e generalização. O coeficiente de determinação também foi semelhante nas duas abordagens, indicando que o modelo explica cerca de 28%, de acordo com a tabela III da variabilidade do redshift a partir dos preditores disponíveis.

TABLE II
COMPARAÇÃO DE RMSE ENTRE OLS DO ZERO E OLS (SKLEARN)

Modelo	RMSE Treino	RMSE Teste
OLS (do zero)	0.619492	0.620604
OLS (sklearn)	0.619492	0.620604

TABLE III
COMPARAÇÃO DE R^2 ENTRE OLS DO ZERO E OLS (SKLEARN)

Modelo	R^2 Treino	R^2 Teste
OLS (do zero)	0.280937	0.279839
OLS (sklearn)	0.280937	0.279839

A validação cruzada em 10 folds, com os resultado demostrados na tabelaIV, reforça esses resultados: tanto a implementação manual quanto a versão com scikit-learn apresentaram RMSE médio próximo de 0.6195, com desvios relativamente pequenos entre os folds. Os valores de R^2 médios na Cross Validation também foram muito próximos (cerca de 0.28), demonstrando estabilidade do modelo ao longo das partições do conjunto de treinamento. A pequena diferença numérica entre as duas abordagens pode ser atribuída a detalhes internos de implementação, como métodos de inversão matricial e otimizações de precisão, mas não representa divergências substantivas no comportamento do modelo.

TABLE IV
RESULTADOS DA VALIDAÇÃO CRUZADA (10-FOLD) PARA OLS

Modelo	RMSE médio (± DP)	R^2 médio (± DP)
OLS (do zero)	0.619501 ± 0.011482	0.280569 ± 0.007124
OLS (sklearn)	0.619550 ± 0.007352	0.280645 ± 0.005428

- Comparando o resultado das duas implementações, com os valores explicitados na tabela IV Obtemos um erro percentual de $0,0079\%$ para o RMSE médio e $0,0270\%$ para o R^2 médio.

De forma geral, os resultados indicam que o OLS fornece um ajuste limitado, capturando parte da estrutura dos dados, mas deixando uma proporção substancial da variação do redshift não explicada. Isso sugere que relações potencialmente não lineares ou interações entre as variáveis podem estar presentes e não são totalmente capturadas por um modelo linear simples.

G. Ridge

Os resultados obtidos para a Regressão Ridge mostram que tanto a implementação do zero quanto a versão da biblioteca scikit-learn apresentaram desempenho praticamente idêntico. Em termos de erro de predição, os valores de RMSE, explicados na tabela V, no treino e no teste são muito próximos aos observados no OLS, indicando que a regularização L2 não

alterou substancialmente o ajuste do modelo — o que sugere baixa multicolinearidade ou baixa variância nos coeficientes.

TABLE V
COMPARAÇÃO DOS VALORES DE RMSE PARA A REGRESSÃO RIDGE

Modelo	RMSE Treino	RMSE Teste
Ridge (do zero)	0.619493	0.620582
Ridge (sklearn)	0.619493	0.620582

Na tabela VI coeficiente de determinação também permaneceu estável entre as versões, com R^2 aproximadamente 0.281 no treino e 0.280 no teste, reforçando que o termo de penalização não levou a melhorias significativas em termos de capacidade preditiva.

TABLE VI
COMPARAÇÃO DOS VALORES DE R² PARA A REGRESSÃO RIDGE

Modelo	R² Treino	R² Teste
Ridge (do zero)	0.280937	0.279891
Ridge (sklearn)	0.280937	0.279891

De acordo com a tabela VII validação cruzada (10-fold) confirma esses achados: O Ridge do zero apresentou RMSE médio = 0.619427 e R^2 medio =0.280794, com desvio-padrão igual a zero devido à forma como o código gera os folds. A versão sklearn obteve resultados muito próximos, com RMSE médio = 0.619550 e R^2 = 0.280646

TABLE VII
RESULTADOS DA VALIDAÇÃO CRUZADA (10-FOLD) PARA RIDGE

Modelo	RMSE médio (± DP)	R² médio (± DP)
Ridge (do zero)	0.619427 ± 0.000000	0.280794 ± 0.000000
Ridge (sklearn)	0.619550 ± 0.007356	0.280646 ± 0.005412

- Novamente Comparando o resultado das duas implementações Obtemos um erro percentual de $0,016\%$ para o RMSE médio e $0,036\%$ para o R^2 médio, de acordo com os valores da tabela VII

Para selecionar o melhor valor do parâmetro de regularização λ da Regressão Ridge, foi realizada uma validação cruzada 10-fold variando esse hiperparâmetro em um conjunto predefinido de valores. a figura II-G permite avaliar diretamente como o erro médio de predição se comporta à medida que a regularização aumenta. De forma complementar a figura II-G mostra a proporção da variabilidade explicada em cada cenário. A análise conjunta das duas tabelas permite selecionar o valor de λ que apresenta o menor RSME e maior R^2 garantindo um bom equilíbrio entre viés e variância, que no caso é aproximadamente $\lambda = 100$ se mostrou ser o valor mais adequado.

De maneira geral, os resultados mostram que a regularização não trouxe ganhos expressivos de desempenho, indicando que o modelo linear simples já era suficientemente estável para o conjunto de dados. No entanto, a análise reforça a importância da validação cruzada na escolha do parâmetro λ, garantindo que o modelo não superajuste nem subajuste ao conjunto de treinamento.

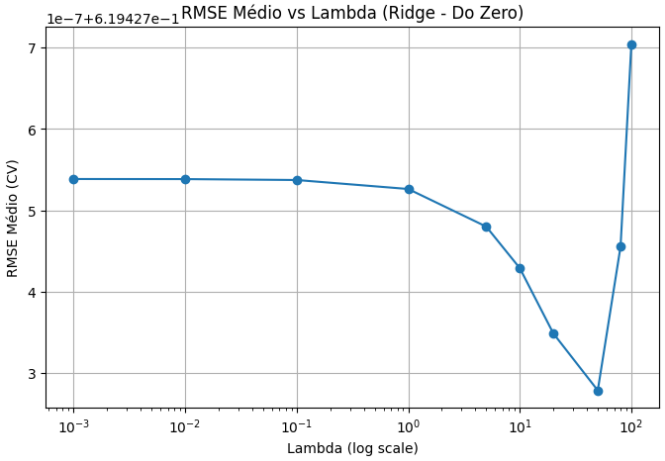

Fig. 1.

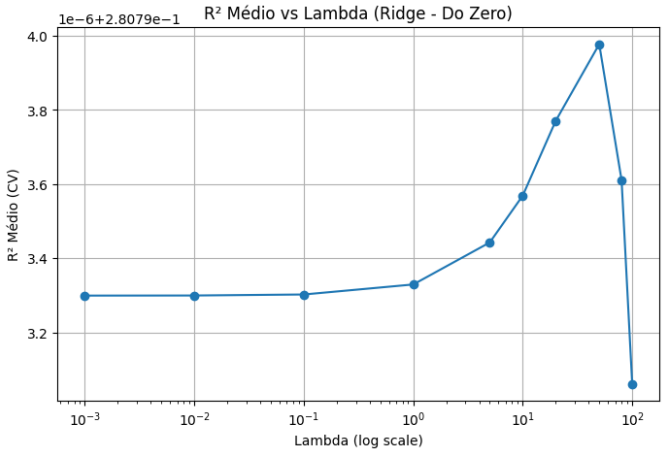

Fig. 2.

H. PCR

A determinação do número ideal de componentes principais foi realizada por validação cruzada 10-fold, avaliando o desempenho do modelo para quantidades variáveis de componentes, para cada valor foram calculados o RSME e R^2 medios cujas tendências foram representadas nos gráficos II-H e II-H

Os gráficos evidenciam que o RMSE diminui rapidamente nos primeiros componentes e se estabiliza após aproximadamente 7 componentes, indicando que a maior parte da variabilidade relevante para prever redshift é capturada logo nas primeiras direções principais. Da mesma forma, o gráfico de R^2 mostra um aumento inicial seguido de estabilização, também por volta de 7 componentes. Após esse ponto, a inclusão de mais componentes não melhora o desempenho e pode, inclusive, introduzir ruído, aumentando a variância do modelo.

Também foi analisada a variância explicada cumulativa dos componentes principais. O gráfico II-H correspondente mostra a fração acumulada da variância total dos preditores capturada pelos primeiros componentes. Observa-se que cerca

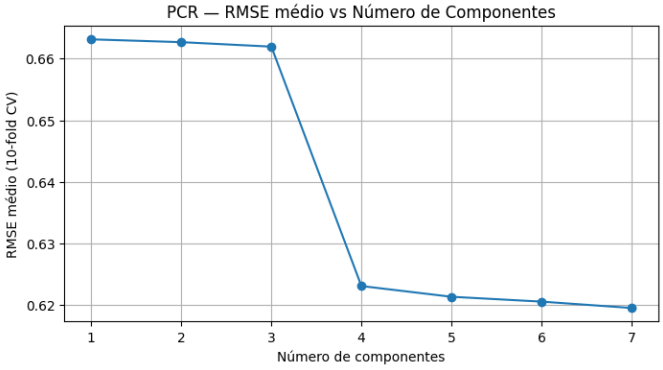

Fig. 3.

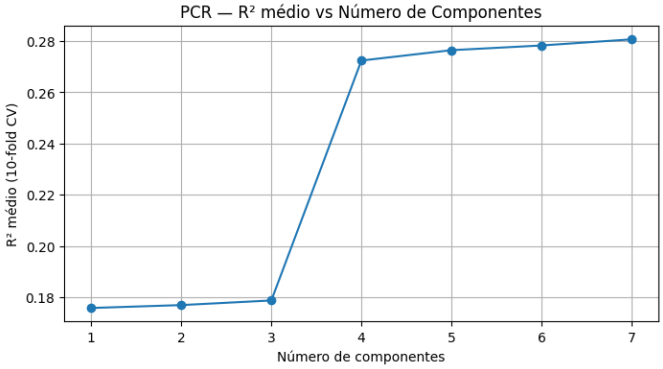

Fig. 4.

de 7 componentes já explicam a maior parte da variabilidade relevante do conjunto de dados, enquanto os componentes adicionais contribuem apenas marginalmente para o aumento da variância explicada.

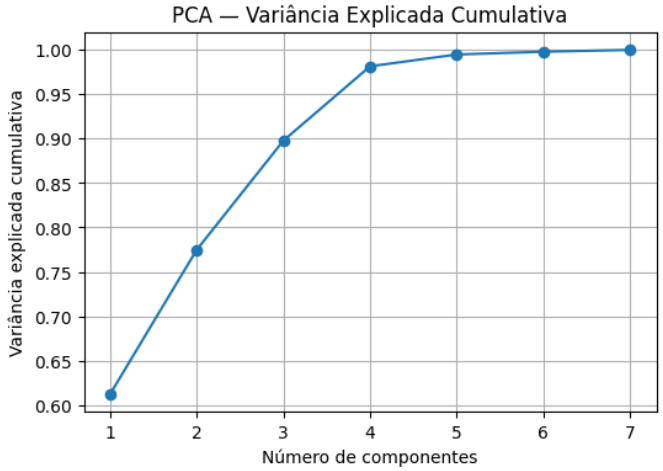

Fig. 5.

Métrica	Treino	Teste	Cross-Validation (10-fold)
RMSE	0.619492	0.620604	0.638908 ± 0.020529
R^2	0.280937	0.279839	0.234210 ± 0.049383

TABLE VIII
RESULTADOS DO MODELO PCR: DESEMPENHO NO TREINO, TESTE E VALIDAÇÃO CRUZADA.

Ao analisar a tabela II-H o modelo de PCR apresentou desempenho levemente inferior ao da regressão OLS e Ridge, mesmo utilizando apenas uma fração da dimensionalidade original indicando que a maior parte da informação relevante presente nos preditores originais está concentrada nos primeiros componentes principais. Isso também sugere que não há uma relação não linear forte nem uma multicolinearidade severa que prejudique o desempenho do OLS.

Como esperado, o erro médio da validação é ligeiramente maior do que o erro obtido no conjunto de teste, já que cada fold utiliza menos dados para o treinamento.

No geral, o PCR mostrou desempenho consistente, confirmando que a redução de dimensionalidade não compromete a qualidade do modelo e reforçando que os componentes principais capturam adequadamente a variabilidade das variáveis originais

I. Rede Neural

A rede neural Multilayer Perceptron (MLP) apresentou desempenho superior aos modelos lineares anteriores. Os resultados mostram que o modelo consegue capturar relações não lineares entre os preditores e o redshift, produzindo métricas significativamente melhores.

Na tabela II-I observa-se que no conjunto de treinamento, o modelo obteve um RMSE = 0.5102 e R^2 = 0.5123, enquanto no conjunto de teste o desempenho permaneceu próximo, com RMSE = 0.5346 e R^2 = 0.4656. Essa proximidade entre treino e teste indica boa generalização e ausência de overfitting relevante, em parte devido ao uso de early stopping durante o treinamento.

TABLE IX
RESULTADOS DA REDE NEURAL MLP

Métrica	Treino	Teste	Cross-Validation (10-fold)
RMSE	0.5102	0.5346	0.5327 ± 0.0129
R²	0.5123	0.4656	0.4680 ± 0.0193

Além disso, foi realizada uma validação cruzada 10-fold para avaliar a estabilidade do modelo. Os resultados foram consistentes, com RMSE médio = 0.5327 e R^2 médio = 0.4680 A baixa variabilidade entre os folds mostra que o desempenho da rede neural é estável em diferentes partições do conjunto de treinamento. Por fim, a curva de aprendizado no gráfico II-I apresenta redução suave da função de loss até a convergência, o que indica um processo de otimização estável.

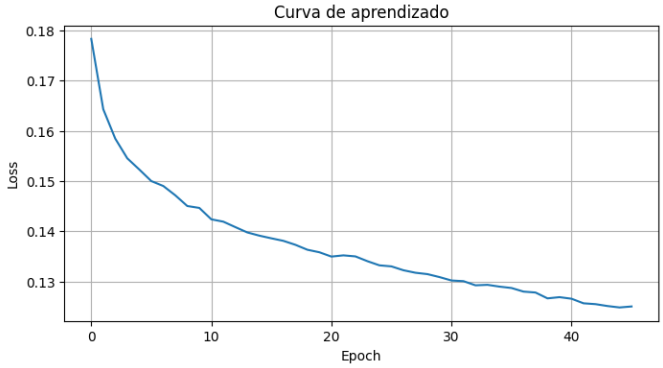

Fig. 6.

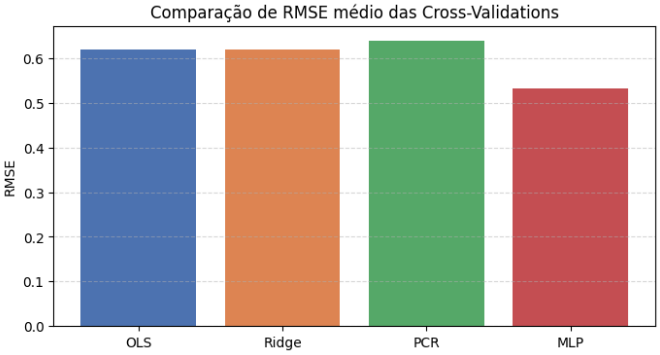

Fig. 7.

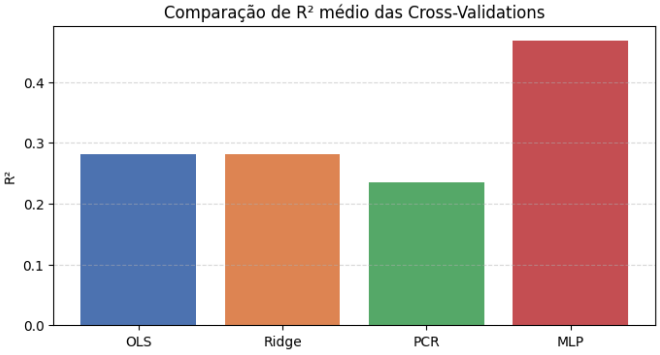

Fig. 8.

J. Comparação geral entre as regressões

A análise dos resultados obtidos pelas quatro abordagens — OLS, Ridge, PCR e Rede Neural (MLP) — evidencia diferenças importantes quanto ao poder preditivo e à capacidade de generalização dos modelos aplicados ao problema de estimativa de redshift. Os gráficos II-J e II-J apresentam um comparativo entre as Cross Validation de todos os modelos utilizados neste estudo, em casos de mais de uma implementação foi feita a media entre os 2 dados correspondentes.

Levando em consideração todos os dados apresentados durante este relatório, nota-se que métodos lineares clássicos, OLS e Ridge, apresentaram desempenhos praticamente idênticos tanto nas métricas de treino e teste quanto na validação cruzada. Por outro lado, o método PCR apresentou o pior desempenho entre os quatro modelos. Apesar de reduzir a dimensionalidade e capturar a variância máxima dos preditores.

A Rede Neural MLP foi o modelo que apresentou o melhor resultado geral, tanto no conjunto de teste quanto na validação cruzada, ela superou de forma consistente todos os métodos lineares. Esse ganho pode ser explicado pela capacidade da rede neural de capturar relações não lineares entre os atributos fotométricos e o redshift, explorando padrões que modelos lineares não conseguem representar. Ainda assim, mesmo apresentando melhor desempenho, o resultado indica que a natureza do dataset é complexa e que a relação entre os atributos e o redshift continua limitada, o que é consistente com estudos astronômicos reais sobre predição de redshift apenas com dados fotométricos.

De forma geral, a comparação mostra que os modelos lineares capturam apenas uma fração modesta da variabilidade, o PCR perde desempenho devido à projeção, e a MLP se destaca pela flexibilidade e maior capacidade de modelagem, sendo a abordagem mais adequada entre as testadas para este problema.

REFERENCES

[1] G. Kauffmann and M. Haehnelt, “A unified model for the evolution of galaxies and quasars”Mon. Not. R. Astron. Soc., vol. 311, pp. 576–588, 2000.

[2] E. M. Burbidge, G. R. Burbidge, W. A. Fowler, and F. Hoyle, “Synthesis of the elements in stars,” Rev. Mod. Phys., vol. 29, no. 4, pp. 547–650, Oct. 1957

[3] Sloan Digital Sky Survey, “Instruments,”SDSS, 2025. [Online]. Available: https://www.sdss.org/instruments/.

[4] SILVEIRA, F. V. C.; MESQUITA, C. V. S.; SOUSA, T. S. O Céu em Dados: Análise exploratória em dados de Galáxias, Estrelas e Quasares. 2025. Disponível em: O Céu em Dados: Análise exploratória em dados de Galáxias, Estrelas e Quasares.pdf.