CTA200 Project

Matteo Moretti

May 2025

Introduction

Redshift refers to the phenomenon where light from a distant source is observed at longer wavelengths (shifted toward the red end of the spectrum) compared to when it was emitted. This occurs because as light travels through the expanding Universe, the stretching of space itself causes the wavelength of the light to increase, resulting in the observed redshift. With redshift, you can determine the distance of the light source, which proves to be extremely important in many fields in cosmology.

One common way to measure redshift is using spectroscopy. By capturing the spectrum of an object, you can calculate the shift of emission lines to get an accurate estimation of redshift ($\delta z \approx 10^{-3}$). However, this method is time-consuming and expensive; it simply cannot keep up with the magnitude of data being collected by new projects [1].

There is another way to measure redshift. Broad-band photometry, which measures flux through different optical bands, is appealing because it is significantly less expensive, and can measure more objects in a given time period than using spectroscopy. The tradeoff is a decrease in accuracy ($\delta z \approx 10^{-1}$). Although there are many methods for estimating photo-z, the two most common approaches are template fitting and machine learning-based methods. We will be focusing on machine learning-based methods [2].

Specifically, this project uses FlexZBoost, an algorithm for conditional density estimation that is well-suited for probabilistic photo-z prediction. FlexZBoost builds on the FlexCode framework by combining basis function expansions with gradient boosting (XGBoost) regression. The FlexZBoost model represents the conditional density $\hat{f}(z \mid x)$ as a weighted sum of basis functions:

$$\hat{f}(z \mid x) = \sum_{j=1}^{J} \beta_j(x)\, \phi_j(z), \tag{1}$$

where $\phi_j(z)$ are predefined basis functions and $\beta_j(x)$ are the corresponding coefficients (or weights) predicted by the regression model based on the photometric features x, as shown in Equation 1. By training on a labeled dataset, this approach allows us to generate full redshift probability density functions (PDFs) for each object. The model's performance is assessed using the Probability Integral Transform (PIT), which evaluates the calibration of these probabilistic predictions.

Preparing the Data

For this project, we will use data from the LSST DEST Data Challenge 1.

Feature Engineering

We performed feature engineering by deriving colors from the raw photometric magnitudes and propagating their corresponding uncertainties. These features

were constructed for both the training and testing data sets.

Colors

Colors are defined as the differences between magnitudes in adjacent photometric bands. Specifically, we computed the following:

$$\begin{aligned}
\texttt{ug} &= U - G \\
\texttt{gr} &= G - R \\
\texttt{ri} &= R - I \\
\texttt{iz} &= I - Z \\
\texttt{zy} &= Z - Y
\end{aligned}$$

These colors define the shape of the spectral energy distribution (SED), which are useful for seeing the shift in spectral features.

Color Uncertainties

To account for measurement noise in the observed magnitudes, we propagated the uncertainties under the assumption of independent, normally distributed errors:

$$\sigma_{X-Y} = \sqrt{\sigma_X^2 + \sigma_Y^2} \tag{2}$$

The following error terms were computed:

$$\begin{aligned}
\texttt{ug_err} &= \sqrt{\texttt{UERR}^2 + \texttt{GERR}^2} \\
\texttt{gr_err} &= \sqrt{\texttt{GERR}^2 + \texttt{RERR}^2} \\
\texttt{ri_err} &= \sqrt{\texttt{RERR}^2 + \texttt{IERR}^2} \\
\texttt{iz_err} &= \sqrt{\texttt{IERR}^2 + \texttt{ZERR}^2} \\
\texttt{zy_err} &= \sqrt{\texttt{ZERR}^2 + \texttt{YERR}^2}
\end{aligned}$$

Incorporating both colors and their associated uncertainties enhances the input feature space of the model by capturing not only the relative spectral information in the photometric gradients but also the corresponding measurement uncertainties.

The input features were normalized to ensure that all variables contribute comparably to the model training process, preventing features with larger numerical ranges from dominating the learning algorithm.

Generating PDFs

The first goal of the project is to generate photo-z probability density functions (PDFs). We take this approach (as opposed to a traditional point estimate) to prevent biases that arise from uncertainties in the training data.

To begin, we performed a comprehensive grid search over XGBoost's regression parameters, as well as the basis function settings used in the FlexCode framework. For every configuration of parameters, the model was trained on the test set and evaluated on the cross-validation set. This was done using a custom loss function `cde_loss` that compares the predicted conditional density estimates with the true redshift values. The configuration with the lowest loss (-8.93) was set as the ideal set of parameters.

Using the selected parameter configuration, we generated redshift probability density functions for the test set. Figure 1 presents representative examples of these PDFs together with the corresponding true redshift values for visual comparison.

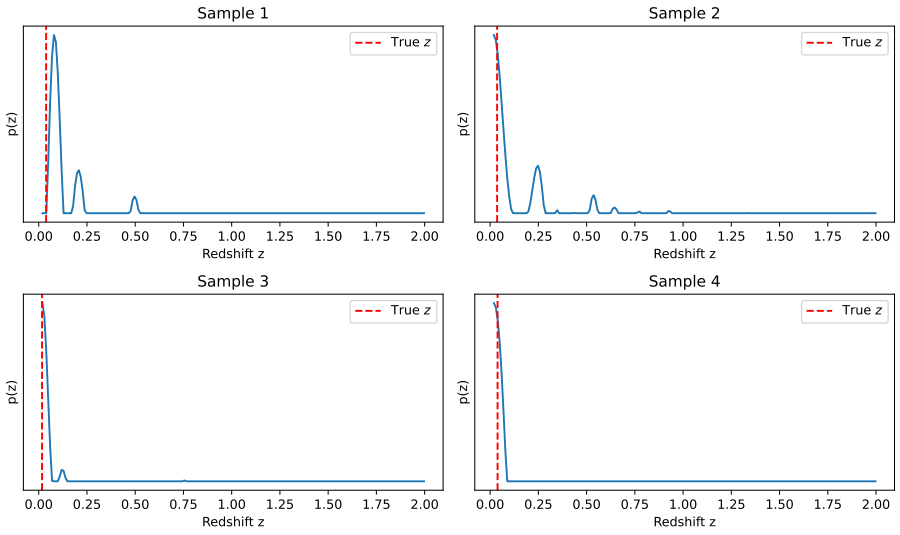

Figure 1: Examples of redshift probability density functions (PDFs) predicted by the model, shown in blue, with the true redshift values indicated by red vertical lines.

Evaluating the Model

To evaluate the model, we use a Probability Integral Transform (PIT). To do this, we create a histogram where the value for each test point is the integral of its predicted PDF up to the true redshift, as shown in Equation 3.

$$\int_0^{z_{true}} \hat{f}(z \mid x) dz \tag{3}$$

If the predicted distributions are well-calibrated, the PIT values will follow a uniform distribution. Deviations from uniformity indicate issues such as overconfidence (U-shaped), underconfidence (inverted U), or systematic bias (skew).

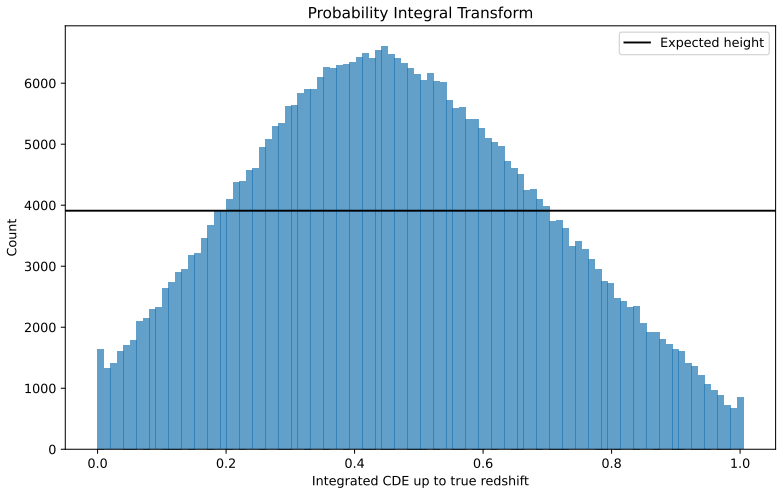

Figure 2: Probability Integral Transform (PIT) histogram. The blue bars represent the PIT values from the data, and the black line indicates the expected uniform distribution.

This was done using a trapezoidal Riemann sum, `numpy.trapz` for each PDF in the catalog. As seen in Figure 2, the model exhibits underconfidence. The U-shape suggests that the predicted distributions are too broad, often assigning lower probabilities to the true redshift values.

References

[1] Schmidt, S. J., Malz, A. I., & Soo, J. Y. (2021). *FlexZBoost: Flexible Photometric Redshift PDFs via Boosted Trees*. Frontiers in Astronomy and Space Sciences, 8, 658229.
https://www.frontiersin.org/articles/10.3389/fspas.2021.658229/full

[2] Izbicki, R., Lee, A. B., & Freeman, P. E. (2020). *Photometric redshift estimation: an end-to-end probabilistic framework*. arXiv:2001.03621.