Draft: Calibration of DESI BGS readsfiht measurements using Kernel methods in Machine Learning

by

Sergio David Lobo Bolaño

Undergraduate Monograph

Submitted to the Department of Physics
in fulfillment of the requirements for
the degree of

Bachelor of Science in Physics

Advisor

Jaime Ernesto Forero-Romero

Department of Physics
UNIVERSIDAD DE LOS ANDES
COLOMBIA
November 19, 2018

Contents

Chapter 1

Introduction

TODO Introduction and motivation of the project.

1.1 Objectives

1.1.1 General Objective

- Determine the relation between the redshift simulated mesurements by DESI and the intrinsic (simulated) redshift of BGS galaxies, to calibrate the instrument for real-world use using kernel method of machine learning

1.1.2 Specific Objective

- Characterize the dataset using a simple statistical and graphical procedure to select the set of meaningful features as input to the machine learning algorithms.
- Determine the set of computational parameters such as memory requirement, number of processors per node and size of dataset that performs the best on the cluster of the university restricted to constraints of resources, execution time and waiting time in queue.
- Train and adjust the hyper-parameter of the models by using grid-search and cross validation.
- Select the best model based on performance on unseen data and model simplicity.

1.2 Context

TODO Important concepts and definition from the DESI project and Machine learning

1.2.1 DESI Project

TODO Description of the Desi project and instrument, how the results are simulated.

1.2.2 Machine Learning

TODO Supervised Learning, regression and classification, principal method of ML:

Kernel Methods

TODO Introduction, mathematics, and parameters of the model.

Neural Networks

TODO Introduction, mathematics, and parameters of the model.

conclusions about the complexity of the neural networks.

Chapter 2

Methodology

2.1 Introduction

This chapter explain the procedure to achieve the objectives declared in the previous chapter. Each objective consist of a series of activities that produce an outcome, each one contributing to reach the general objective of this monograph. First, a description of the data pre-processing, then the selection of computer parameters and finally, a description of the methodology for training and testing the models.

2.2 Data analysis and pre-processing

The dataset consists of two files, one with the redshift measurement of DESI (TAR file) and the other with simulated true redshifts (TRUTH file). Apart from this, the files also contain several variables related to the simulations parameters, the type of target, the observation conditions, etc. Figure 2.1 shows the general step-by-step method to understand the data. In the first step, the two files have to be linked by the variable TARGETID, this variable represent the identity of a target through the whole pipeline of the experiments (observation and measurements). After this, the next step is to discriminate by target type and observe the behavior of the other variables in the files. From this one can neglect some basic variables that give no relevant information to the problem. A second analysis can be executed with more details, to see if the variables hold any relation with the redshift measurement, using histograms and approximating distribution. This is done to select which variables (or features) can be used to explain the difference between the true redshift of a galaxy and the redshift measured by DESI. The details of this can be found in Chapter 3. After the features are selected, they will be scaled so that the means are equal to zero and the variance to one.

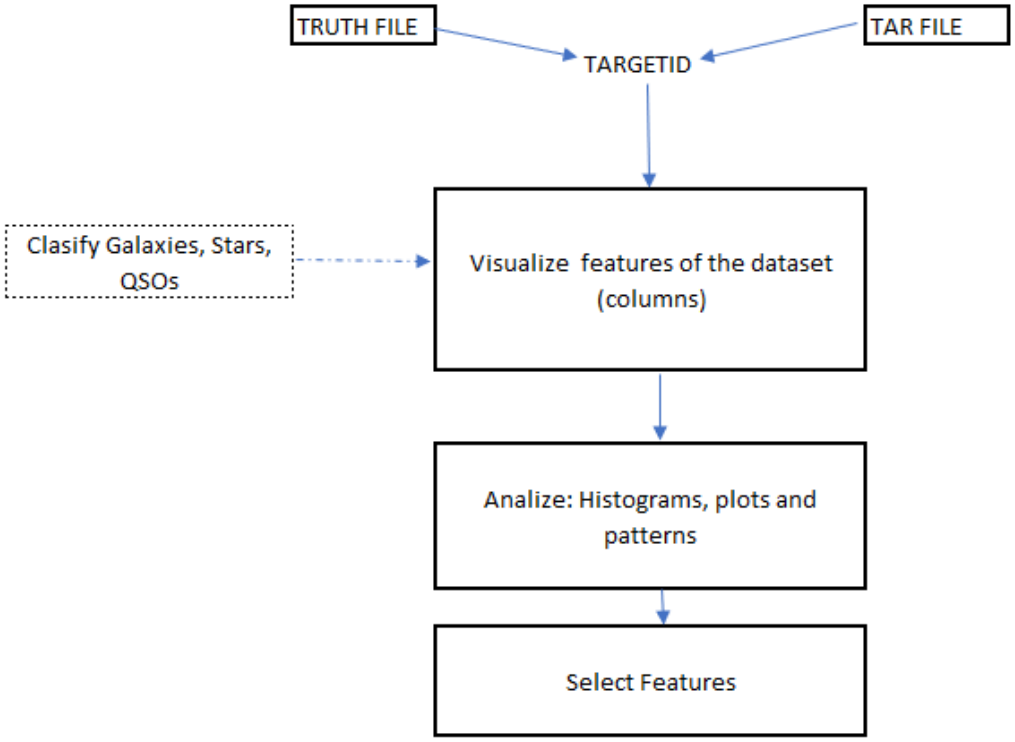


Figure 2.1: General methodology to understand the dataset and extract valuable features.

2.3 High performance computing analysis

The training time if the model depends on the size of the training set, however, since it is going to be executed in parallel in the computing cluster of the University, the training time will also depend on the resources of the hardware. For example, the memory of the machine and the number of processor in which the code will be parallelized, as well as the number of processor that the computer (or node) actually has. To see what is the best combination of parameters to train the models in a reasonable amount of time up to a reasonable accuracy, it will be necessary to restrict also the size of the dataset used for training.

Figure 2.2 shows the step to accomplish this. First, select the model and architecture (grid search / cross validation, explained next) and the dataset size, in order to see how the training time scales with size, it is necessary to select different sizes, in this case, 100, 1000, 5000, and 10000. Then, the algorithm have to be run in certain number of processors, and with certain RAM memory,in this case, we will try 1, 4, 8 and 16 processors, and 16, 32, 64 and 128 GB of memory. The expected result is number of parameters to run the complete training algorithm on a big dataset.

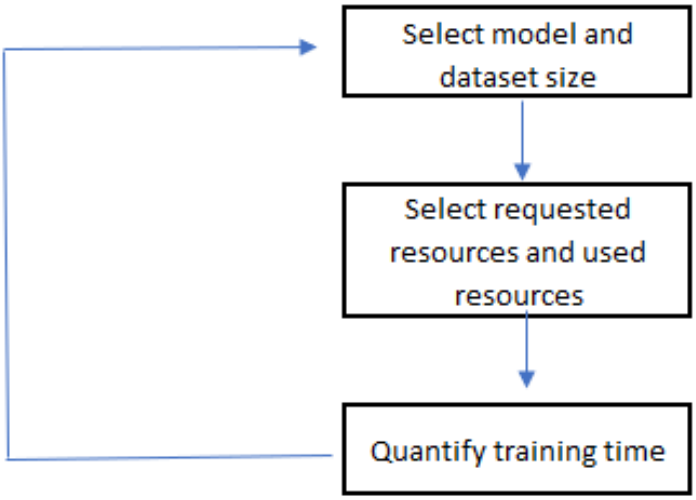


Figure 2.2: Sequence of step to gather enough information to understend the relation between computation time, memory, dataset size and processors.

2.4 Machine Learning

Once we understand the scalability of the training time with respect to hardware variables, it is time to train and test the models. For this the dataset will be divided in a training set, corresponding to 75% of the dataset and a test set (the rest 25 %) as shown in Figure 2.3. After this, according to the results of the previous section, the training set will be split in subsets of specific size, this subsets will be again divided into a development and evaluation sets, to apply to them the grid search and cross validation. This two methods are used to select the hyper-parameters of the models and to quantify the error of the algorithms.

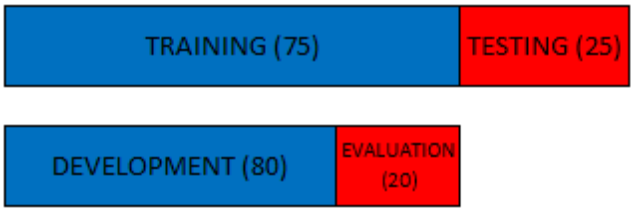


Figure 2.3: Divition of the dataset in train and test. The train set is divided in development-evaluation subsets.

2.4.1 Training

For training, we will use grid search and cross validation to select the hyper-parameters of the models. The grid search method correspond to train the model with different combinations of parameters as in going over a grid, and search for the combinations of parameters that give the best results. This is used alongside cross validation, which

consist on training the model several times by randomly assigning the development and the evaluation sets. Therefore, it reports the mean and standard error of the metrics of the trained models.

2.4.2 Testing

Once the models are trained, they will be test on the unseen 25% of the data to check its generalization ability. This means, that we need to see whether the model truly 'learns' to predict redshifts or just memorized the training dataset. To select the best performing algorithm on the test dataset, the criterion will be the r^2 measure and the simplicity of the model.

In general, the methodology for this project consist in analysing the complete dataset and selecting the valuable features than can be used to predict a correct redshift, finding the scalability of the training time with respect to the size of the dataset and the cluster resources, an finally, training an evaluating the machine learning models.

Chapter 3

Data Analysis and Pre-processing

3.1 Data descripcion

The dataset used consist of two FITS files corresponding to the simulated observations of the DESI instrument and the "truth" data from cosmological simulations. Each data file has a key column called TARGETID, this variables is the same across all simulations and identifies a particular object in the sky, thus it is needed to relate both datasets.

3.1.1 Simulated expected results - truth file

This file contains the data from the cosmological simulation of the target objects. Thus, this file contains the expected redshift that the instrument *should* measure. The complete list of columns in the file is shown in table 3.1[1]. Since we aim to correct the measurements of redshifts given by the instrument, the variable TRUEZ will be our target variable or "*output*" to the machine learning models. The rest of the information will be used mostly for understanding the dataset but no as part of the ML model, since this information would not be available to the instrument in reality. This dataset contains the redshift of 24.851.543 objects.

3.1.2 Simulated Observations - target file

This file contains the redshift values of the targets as measured by the instrument in simulation. Apart from this, it also contains the columns shown in table 3.2. The different characteristics listed in Table 3.2 are the ones that we will use as input in our machine learning models, since is the data available from the instrument. This dataset contains the redshift of 2.131.896 objects. Since the files don´t have the same amount of points, they were cut to the one with the least (the target file) by linking the rows by its TARGETID.

[1] Additional information on `https://desidatamodel.readthedocs.io/en/stable/`

Name	Description
TARGETID	ID (unique to file and the whole survey)
MOCKID	Mock ID
TRUEZ	True redshift in mock catalog (including peculiar velocity)
TRUESPECTYPE	True object type in mock catalog
MAG	
FLUX_G	DECaLS flux from tractor input (g)
FLUX_R	DECaLS flux from tractor input (r)
FLUX_Z	DECaLS flux from tractor input (z)
FLUX_W1	WISE flux in W1
FLUX_W2	WISE flux in W2
OIIFLUX	Flux in OII line
HBETAFLUX	Flux in Hbeta line
TEFF	Effective Temperature
LOGG	Surface Gravity
FEH	Metallicity

Table 3.1: Columns in the cosmological simulation data file

Name	Description
TARGETID	ID (unique to file and the whole survey)
BRICKNAME	Brick name from tractor input
BRICK_OBJID	OBJID (unique to brick, but not to file)
RA	Right ascension [degrees]
DEC	Declination [degrees]
FLUX_G	DECaLS flux from tractor input (g)
FLUX_R	DECaLS flux from tractor input (r)
FLUX_Z	DECaLS flux from tractor input (z)
FLUX_W1	WISE flux in W1
FLUX_W2	WISE flux in W2
SHAPEEXP_R	Half-light radius of deVaucouleurs model (>0)
SHAPEEXP_E1	Ellipticity parameter e1 of deVaucouleurs model
SHAPEEXP_E2	Ellipticity parameter e2 of deVaucouleurs model
SHAPEDEV_R	Half-light radius of exponential model (>0)
SHAPEDEV_E1	Ellipticity parameter e1 of exponential model
SHAPEDEV_E2	Ellipticity parameter e1 of exponential model
PSFDEPTH_G	PSF-based depth in DECaLS g
PSFDEPTH_R	PSF-based depth in DECaLS r
PSFDEPTH_Z	PSF-based depth in DECaLS z
GALDEPTH_G	Model-based depth in DECaLS g
GALDEPTH_R	Model-based depth in DECaLS r
GALDEPTH_Z	Model-based depth in DECaLS z
MW_TRANSMISSION_G	Milky Way dust transmission in DECaLS g
MW_TRANSMISSION_R	Milky Way dust transmission in DECaLS r
MW_TRANSMISSION_Z	Milky Way dust transmission in DECaLS z
MW_TRANSMISSION_W1	Milky Way transmission in WISE W1
MW_TRANSMISSION_W2	Milky Way transmission in WISE W2
BRICKID	Brick ID from tractor input
DESI_TARGET	DESI (dark time program) target selection bitmask
BGS_TARGET	BGS (bright time program) target selection bitmask
MWS_TARGET	MWS (bright time program) target selection bitmask
HPXPIXEL	HEALPixel containing target.
CHI2	Best fit chi2
COEFF	Redrock template coefficients
Z	Best fit redshift
ZERR	Uncertainty on best fit redshift
ZWARN	Warning flags; 0 is good
SPECTYPE	Spectral type
SUBTYPE	Spectral subtype (maybe blank)
DELTACHI2	Delta(chi2) to next best fit

Table 3.2: Columns in the Simulated Observations data file

3.2 Overview of the dataset

3.2.1 Redshift relations

To understand the dataset, the first thing we need to do is see how the Z and TRUEZ variables behave. In Figure 3.1 we see three distinct regions: a 45-degree line that correspond to the redshifts measurements of DESI that are very close to the expected 'real' value, a square region where the data seems to scatter randomly except for some line groupings, and a third region of horizontal line near True Z = 0. The task is therefore to dissolve the square region and have all the points along the diagonal line.

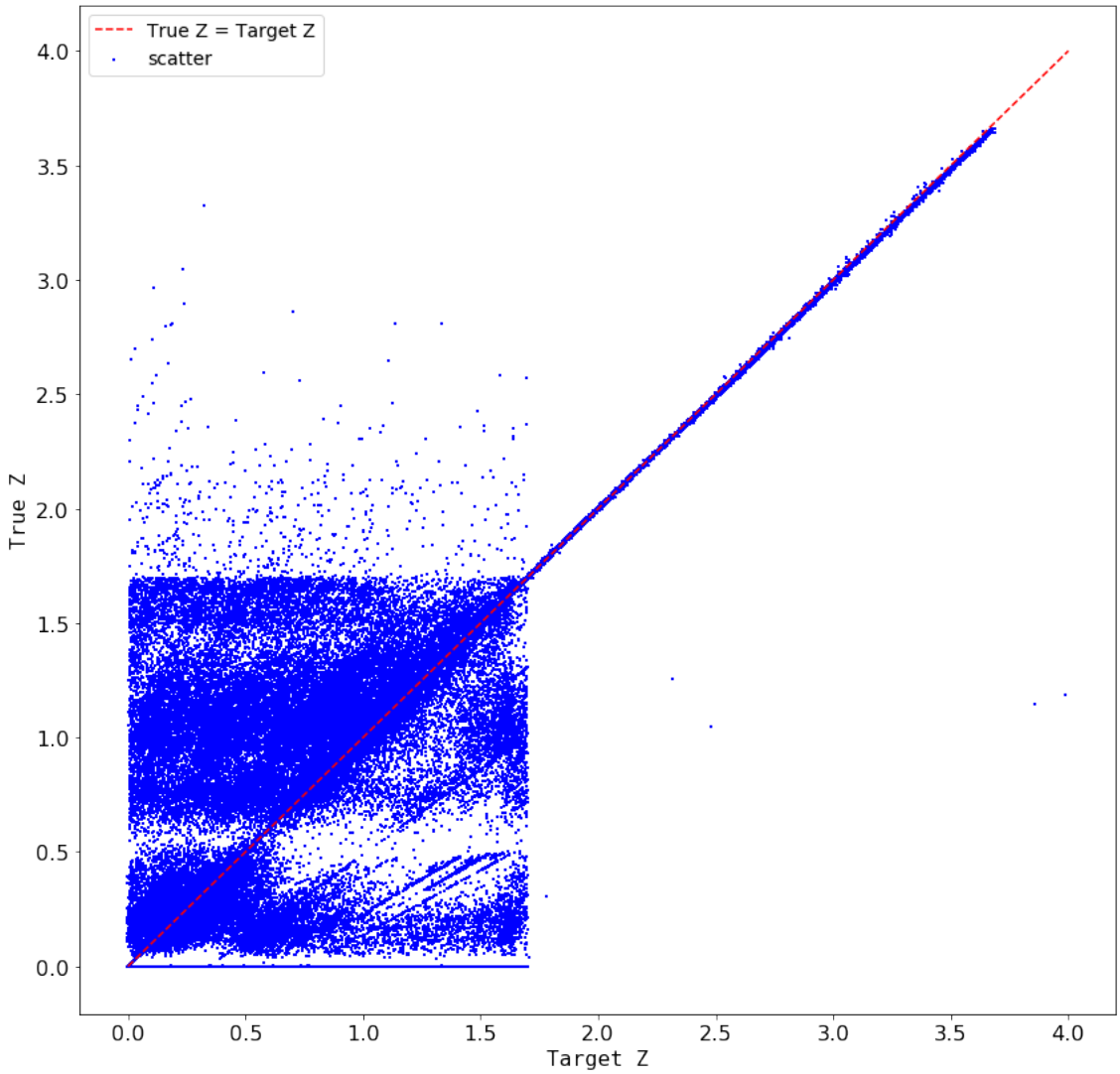
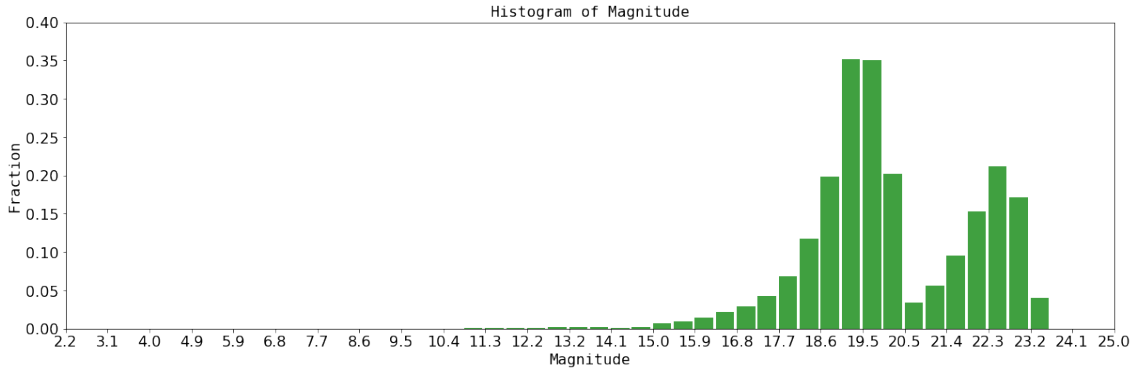
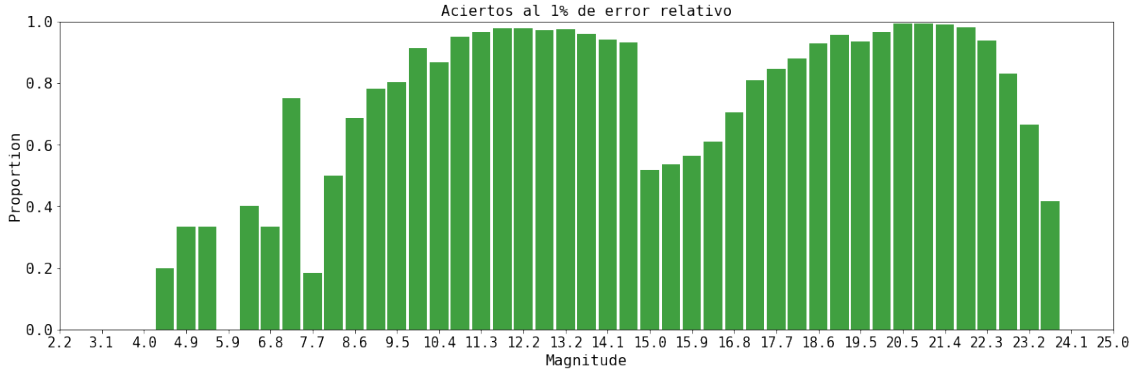


Figure 3.1: Relation between the 'observed' readshift - target Z, and the 'generated' redshift - True Z.

Now, cutting along the MAG (magnitude) variable, it is possible to see the distribution of magnitudes of the gathered data an see if there is any relation with the square region. Figure 3.2a shows this distribution and Figure 3.2b shows the fraction redshifts of each bin that is within a 1% error of the true value. We can see that the majority of the redshift is near its true value, however some valleys indicate that some regions (for example between



(a)



(b)

Figure 3.2: (a) Distribution of magnitudes in the dataset. (b) Proportion of the number of measured redshifts that are within the 1% error of their corresponding TRUEZ

15 and 17 in magnitude) are not that good. For simplicity, from now on we will refer to the **simulated expected redshift** as TRUEZ (the output to the ML models) and the **simulated observations of the targets** as Z (the input to the models) or TARZ.

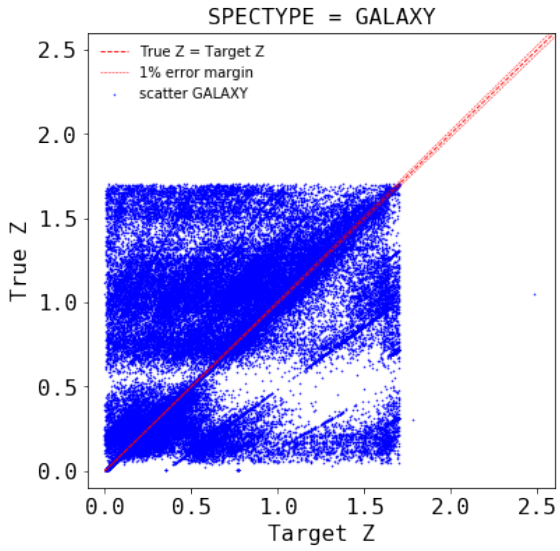
3.2.2 Spectral types

From figure 3.2 it is possible to infer that the different magnitudes of the targets can be related to the different regions on figure 3.1, and this difference in magnitude is also related to the SPECTYPE of each target. The spectral type of the data in the tar-file is distributed as shown in Table 3.3. Near 85% of the dataset is composed of Galaxy-type objects, for this reason, this is the most interesting class.

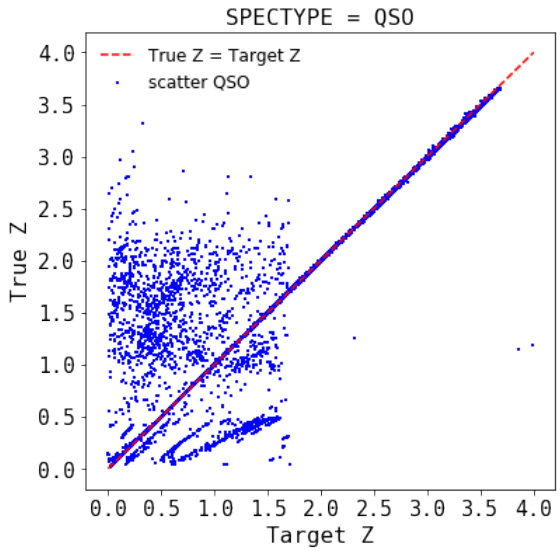
SPECTYPE	N samples	% of dataset
Galaxy	1796213	84.25
QSO	194319	9.11
Star	141364	6.63
Total	2131896	100

Table 3.3: Spectral type distribution of the tar file.

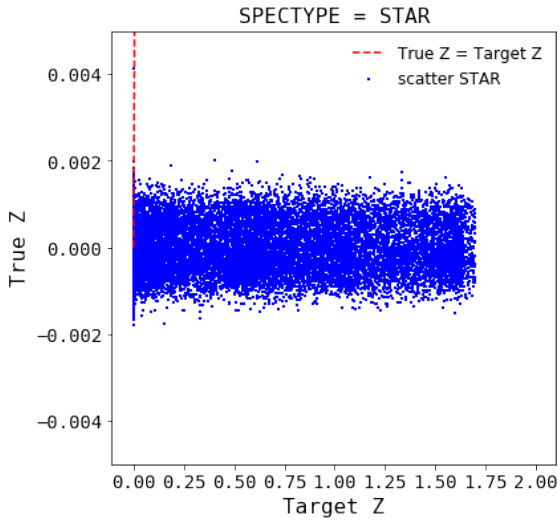
The redshift relations similar to Figure 3.1 discriminated by SPECTYPE are shown in



(a)



(b)



(c)

Figure 3.3: Redshift relation for (a) Galaxy-type objects, (b) QSO-type objects, and (c) Star-type objects

Figure 3.3, where the three regions in Figure 3.1 seems to be related to each spectral type. The GALAXY objects are distributed along a square and there are lines formed at angles different than 45, which means that there is a relation between the two variable but 'out of calibration'. It is worth mentioning that Figure 3.3a can be deceiving, because 96.34% of the Galaxy-type data is within the 1% margin line in the figure, which means that the square is formed only by the 3.66 % of the galaxy data, corresponding to approximately 65.742 measurements, still a lot.

The QSO-type objects in Figure 3.3b correspond to the 45-degree line in Figure 3.1 since the majority of points are along this line, although the same line patter and dispersion of the galaxy-type objects are present, but in least quantity. However, the Star-type in Figure 3.3c is randomly scattered over the TARZ range and correspond to the horizontal line in Figure 3.1. Once again, the most interesting SPECTYPE is GALAXY, because it has the majority of error in TARZ, and also presents different patterns, QSO are already fine and are not a priority while STAR is completely random a represent a small fraction of the whole dataset. From now on we will focus on Galaxy-type objects only.

3.2.3 Galaxy-type objects

Galaxy-type objects are clasified as Bright Galaxy Survey (BGS), Emission Line Galaxies (ELG) and Luminous Red Glaxies (LRG) distributing according to Table 3.4. In this case, the sub-types are more evenly distributed. The redshift relations of each sub-type is shown in Figure 3.4. BGS and ELG follow similar pattern, however ELG is more disperse while BGS is clustered along the TARZ = TRUEZ line, whereas LRG sub-type is already perfect.

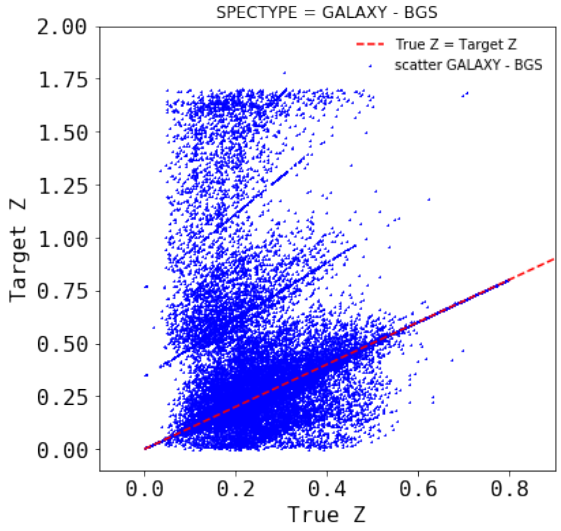
Galaxy subtype	N samples	% of dataset
BGS	889336	49.51
ELG	601847	33.51
LRG	305030	16.98
Total	1796213	100

Table 3.4: Distribution of Galaxy sub-types

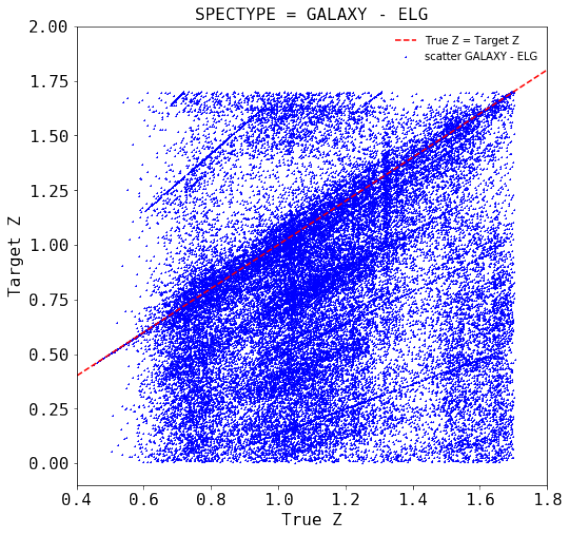
Note that the line structures are more visible in Figure 3.4a than in Figure 3.4b, therefore, to extract the relevant features in the tar dataset that may be related to this structure, we will use the BGS subset and then see if the feature extracted are also useful for the ELG subset. Therefore, from now on we will use only the BGS data subset and proceed to find the relevant features (in Table 3.2)

3.3 Relevant Features

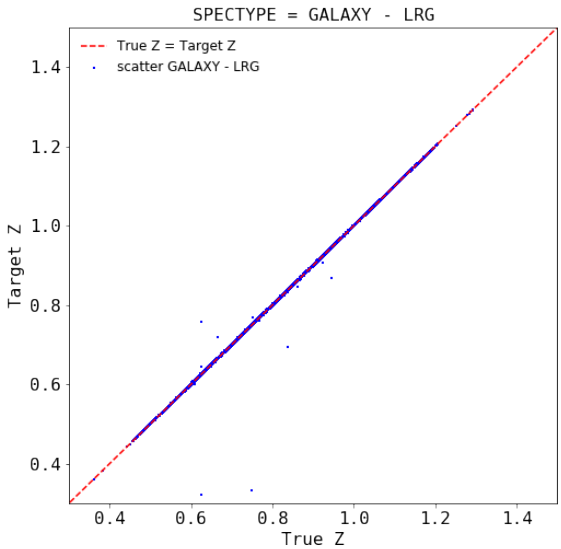
The ability of the instrument for measuring the correct redshift will probably depend on the quality of the fluxes that the fiber optics receive. The fluxes of the objects are related



(a)



(b)



(c)

Figure 3.4: Redshift relation for Galaxy (a) BGS-type objects, (b) ELG-type objects, and (c) LRG-type objects

to its magnitude, as we saw en Figure 3.2, magnitude is related to the correct prediction of the instrument's Z, but since the information of magnitude is known to instrument in form of fluxes, this variables will be explored next. The six fluxes variables are named in Table 3.2. The distributions of the fluxes in the BGS dataset are shown in Figure 3.5.

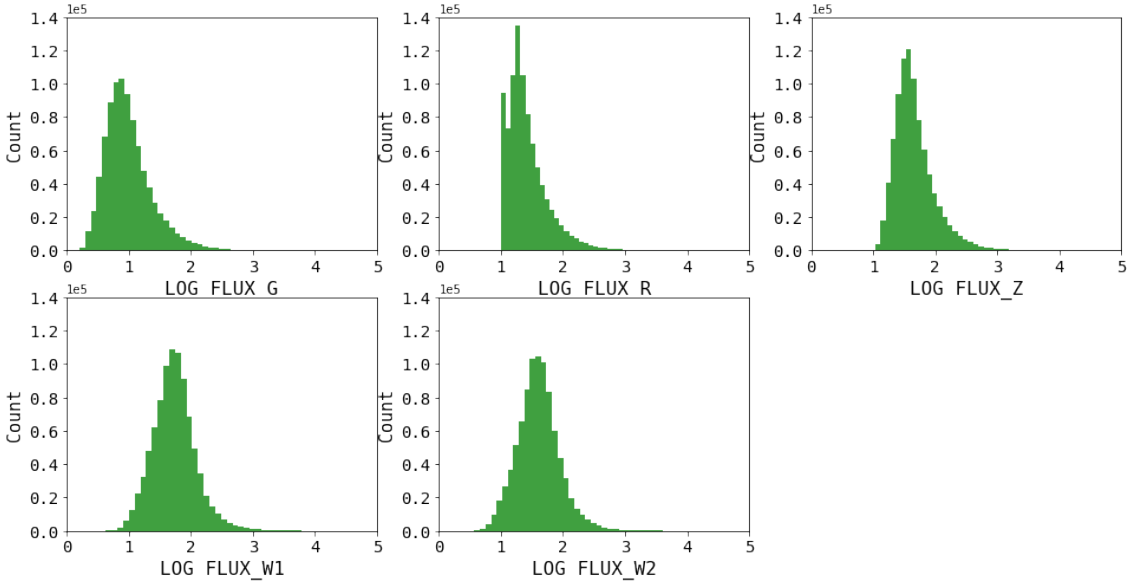
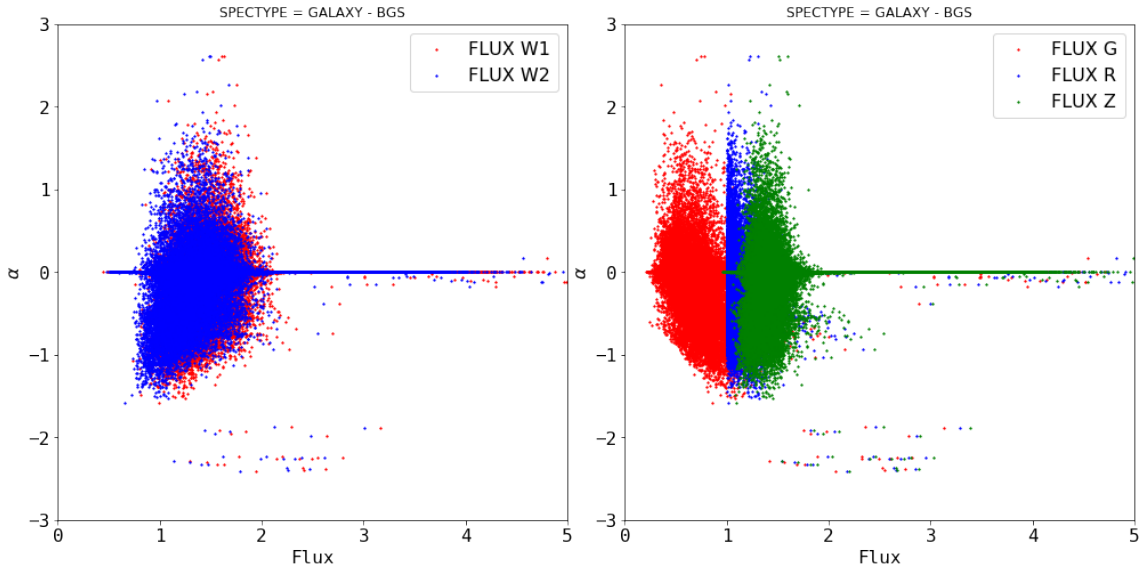


Figure 3.5: Distribution of the flux variable in the BGS subset.

In order to keep track of the relation between TARZ y TRUEZ and see the behavior of the flux variable with respect to the two, we define the following variable

$$\alpha = \frac{TRUEZ}{TARZ}, \tag{3.1}$$

therefore, alpha have a value near 1 when the redshifts are along de 45-degree line.



(a) (b)

Figure 3.6: Relation between fluxes, TRUEZ and TARZ. (a) α as a function of W1 and W2 fluxes. (b)α as a function of G, R and Z fluxes. $\log_{10}$ values presented.

We see in Figure 3.6 that each flux has a similar behaviour, however, they are dispersed in different values and present different minimum values. As the flux magnitude increases, α tends to equal 1, meaning that as the fluxes received by the instrument increases, the redshift measurements are closer to the expected real values. Therefore, the dispersion regions may contain the information necessary for the ML models to learn no predict better redshifts. To see the influence of the dispersion at low fluxes over α, the cuts at different fluxes values in Figure 3.7 show the distribution of α at particular values of the fluxes.

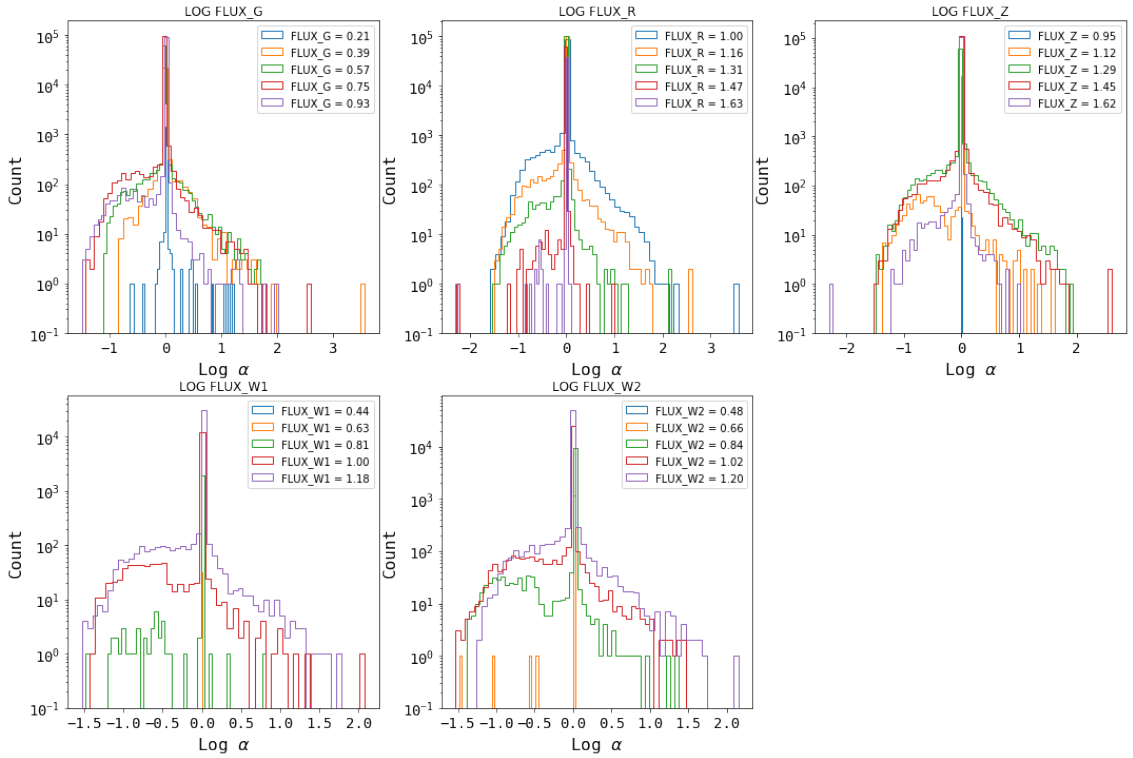


Figure 3.7: Distribution of α at different cuts in the flux variable from figure 3.6

The plots in Figure 3.7 are constructed by taking thin slices at fixed values of the fluxes in Figure 3.6 and constructing the histogram of the points that lie within the slice. The first approximation is to fit the distribution to a Gaussian distribution and estimate its mean and variance. The relation between mean and fluxes is shown in Figure 3.8, were the green points indicate the mean of α for the slice taken at the given flux. The blue line in Figure 3.8 indicates when $TRUEZ = TARZ$, therefore, we see in that for each flux, there are ranges within $\mu(\alpha)$ is not 1, even when the typical error is small. This ranges are around 1 (in log scale) for all the fluxes. Also, note that at very high fluxes, the error in the mean increases, related to the less data available at those values and its high dispersity.

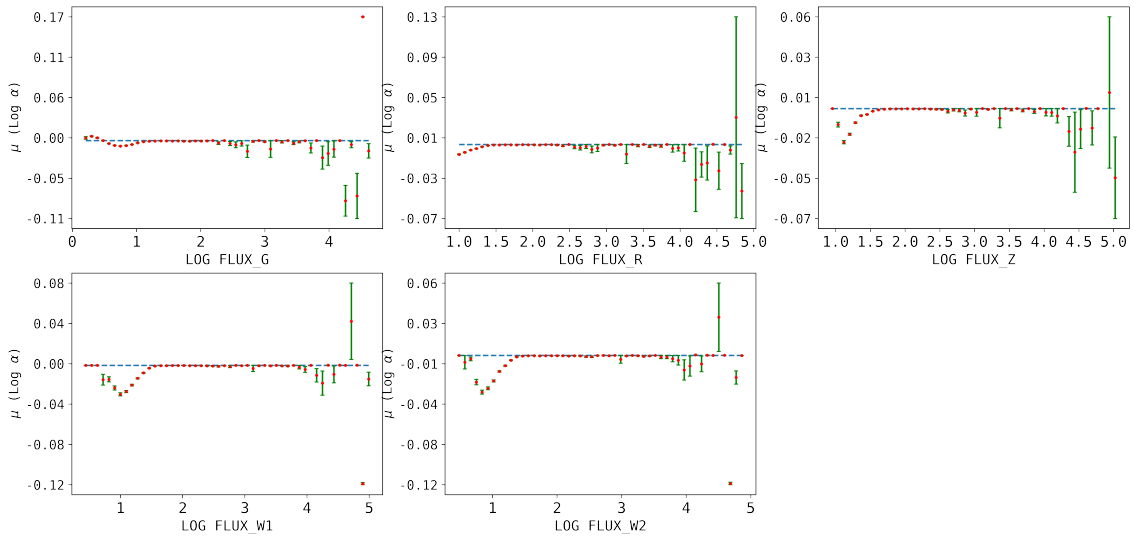


Figure 3.8: Mean (red) and standard error (green) of $\log_{10}(\alpha)$ for every Flux interval (slice).

3.4 Conclusions

The dataset used consist of two sets of simulated observation, the truth-file, with the simulated expected results of redshifts (TRUEZ) from the survey targets, and the target-file with the simulated observations by the instrument of the targets redshits (TARZ). The targest are clasifiedas Galaxies, QSOs and Stars, being Galaxies the most representative type with 84.25% of the dataset. From this, the most interesting subtype of galaxies is the Bright Galaxy Survey (BGS) because of its patterns in redshift and the amount of data (49.51% of the Galaxy type). The results in the following chapter are focused only in the BGS subtype, using as input features for the machine learning algorithms the fluxes and the target redshift, and the output the true redshift.

Chapter 4

Results

4.1 Scalability of the training time of the ML models

Since the BGS dataset contains about 800000 data points, the training of algorithms, model selection and validation can be too demanding, in processing and memory. Therefore, it is necessary to evaluate how the training time scales with the data when running in the High Performance Computing (HPC) cluster of the University, taking account several variables as the number of processor to run in parallel the algorithm and the resources requested to the cluster. In particular, we will focus on the following variables:

$$\begin{aligned} m &= \text{Requested memory to solve the job [Gb]}, \\ n_{jobs} &= \text{Number of jobs to run in parallel}, \\ ppn &= \text{Requested proccesors per node}, \\ n &= \text{size of the dataset used}, \\ M &= \frac{n_{jobs} \times n}{ppn \times m}. \end{aligned}$$

The variable M measures the relation between the requested resources and the used. Figure 4.1 shows how time increases as a function of variable M. Note that M^{-1} can be seen as a 'unitary memory', that is, the memory used per data per processor,

$$M^{-1} = \frac{m}{p_u \times n},$$

where $p_u = n_{jobs}/ppn$ is the fraction of the requested processors that is actually being used by the parallel code. Therefore, when M is big, it means that there is too much data per memory per processor, and the time to train the model is to high. It is worth mentioning that the time were measure for the kernel ridge regression model particularly, using grid search and 3-fold cross validation. Others model may be faster or slower to train, but the behavior should be the same. The time of training for a given M can be aproximated by the relation

$$t = 7M^{1.11}[s]. \tag{4.1}$$

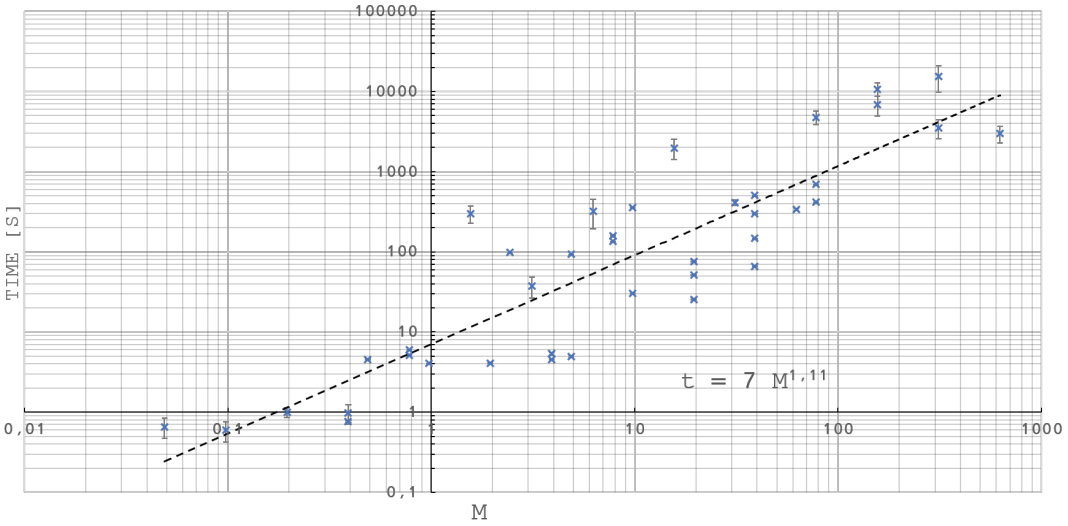


Figure 4.1: Time as a function of requested memory, number of data points and jobs used to train the model.

This results have to be constrained to the waiting time in queue of the cluster, in general, the smaller M, the best time, but that means, using small dataset or increasing the resources, but for the second option, the more resources you request, the longer the queue and waiting time to execute the code. It was fond that a configuration with $n_{jobs} = 16$**,** $ppn = 32$**,** $m = 128Gb$ **and** $n = 100000$, gives good results. n is critical since it determines how well the model will learn, 100000 data points works very well as it will be shown in the next sections.

4.2 Model results

The three model tested consist of a KRR, a SVR and an ensemble model. The original BGS dataset was divided into a 75 - 25 train-test set, from the 75 %, each model was trained on a subset of size 100.000, as mentioned in the previous section. To account for the bias, the same model was trained in tree 100.000 dataset and the results were averaged. The selection of the model hyperparameters was made by using grid search a 3-fold cross validation, using a 80-20 development-evaluation scheme.

4.2.1 Support Vector Regression (SVR) Model

The grid-search and cross-validation on the train-development set use the r^2 measure to select the best parameters.

```
----------------------------------------------------------------------
Model 1:
Best parameters set found on development set:
```

```
{'gamma': 0.1, 'C': 1, 'kernel': 'rbf'}

Grid scores on development set:

0.877 (+/-0.001) for {'gamma': 0.001, 'C': 1, 'kernel': 'rbf'}
0.927 (+/-0.011) for {'gamma': 0.1, 'C': 1, 'kernel': 'rbf'}
0.848 (+/-0.017) for {'gamma': 1, 'C': 1, 'kernel': 'rbf'}
0.896 (+/-0.001) for {'gamma': 0.001, 'C': 10, 'kernel': 'rbf'}
0.924 (+/-0.010) for {'gamma': 0.1, 'C': 10, 'kernel': 'rbf'}
0.870 (+/-0.013) for {'gamma': 1, 'C': 10, 'kernel': 'rbf'}
0.898 (+/-0.006) for {'gamma': 0.001, 'C': 100, 'kernel': 'rbf'}
0.922 (+/-0.007) for {'gamma': 0.1, 'C': 100, 'kernel': 'rbf'}
0.868 (+/-0.020) for {'gamma': 1, 'C': 100, 'kernel': 'rbf'}

r2 score computed on the full evaluation set:

0.928

--------------------------------------------------------------------------------
Model 2:

Best parameters set found on development set:

{'gamma': 0.1, 'C': 1, 'kernel': 'rbf'}

Grid scores on development set:

0.878 (+/-0.002) for {'gamma': 0.001, 'C': 1, 'kernel': 'rbf'}
0.928 (+/-0.007) for {'gamma': 0.1, 'C': 1, 'kernel': 'rbf'}
0.859 (+/-0.020) for {'gamma': 1, 'C': 1, 'kernel': 'rbf'}
0.892 (+/-0.002) for {'gamma': 0.001, 'C': 10, 'kernel': 'rbf'}
0.919 (+/-0.019) for {'gamma': 0.1, 'C': 10, 'kernel': 'rbf'}
0.859 (+/-0.010) for {'gamma': 1, 'C': 10, 'kernel': 'rbf'}
0.902 (+/-0.003) for {'gamma': 0.001, 'C': 100, 'kernel': 'rbf'}
0.909 (+/-0.006) for {'gamma': 0.1, 'C': 100, 'kernel': 'rbf'}
0.861 (+/-0.020) for {'gamma': 1, 'C': 100, 'kernel': 'rbf'}

r2 score computed on the full evaluation set:

0.92
--------------------------------------------------------------------------------
Model 3:
```

```
Best parameters set found on development set:

{'gamma': 0.1, 'C': 1, 'kernel': 'rbf'}

Grid scores on development set:

0.879 (+/-0.003) for {'gamma': 0.001, 'C': 1, 'kernel': 'rbf'}
0.926 (+/-0.019) for {'gamma': 0.1, 'C': 1, 'kernel': 'rbf'}
0.847 (+/-0.008) for {'gamma': 1, 'C': 1, 'kernel': 'rbf'}
0.898 (+/-0.005) for {'gamma': 0.001, 'C': 10, 'kernel': 'rbf'}
0.921 (+/-0.015) for {'gamma': 0.1, 'C': 10, 'kernel': 'rbf'}
0.868 (+/-0.009) for {'gamma': 1, 'C': 10, 'kernel': 'rbf'}
0.901 (+/-0.005) for {'gamma': 0.001, 'C': 100, 'kernel': 'rbf'}
0.891 (+/-0.018) for {'gamma': 0.1, 'C': 100, 'kernel': 'rbf'}
0.853 (+/-0.012) for {'gamma': 1, 'C': 100, 'kernel': 'rbf'}

r2 score computed on the full evaluation set:

0.93
```

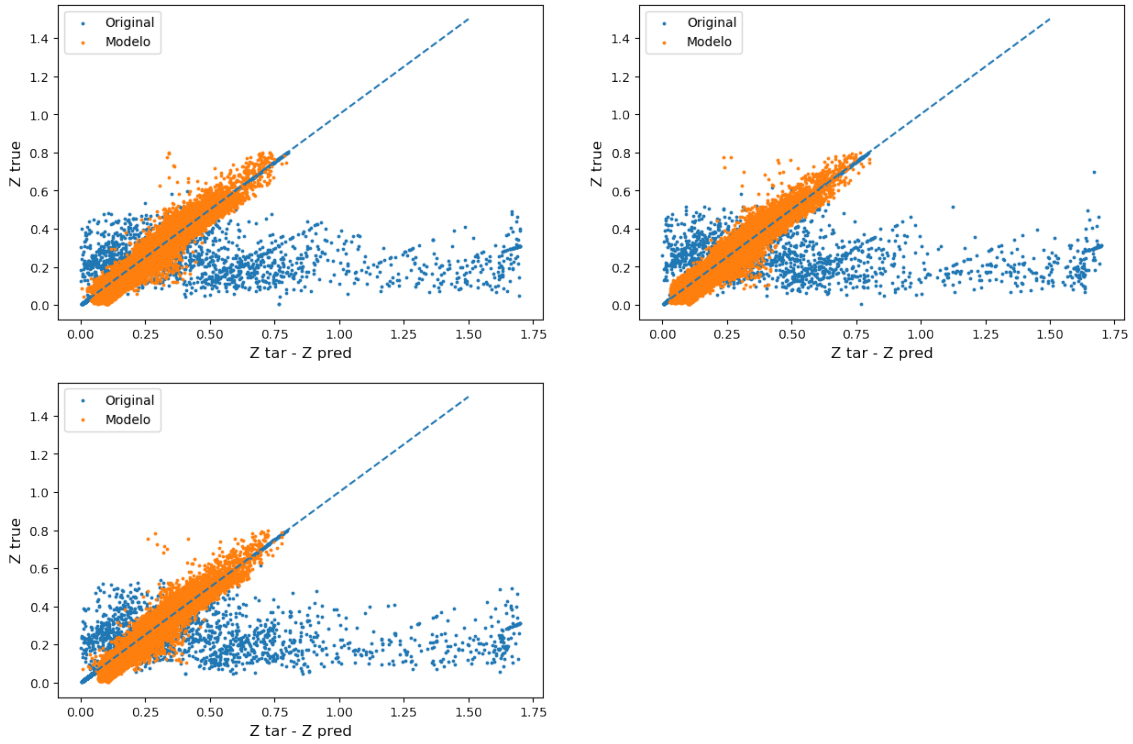


Figure 4.2: Results onthe training set after training the SVR model.

The results on training set of the best models is shown in Figure 4.2. On training, the model shows good results by gathering the blue dots (original data)over the orange dots (the model results after applying it to the training data). The blue scatter is substantially

diminished and the data is clustered along de TRUEZ = TARZ line, However, the data is not entirely over the line, but in a wide range. The result on the evaluation set for each model is show in Figure 4.3. Figure 4.3 shows how the model generalize to unseen data.

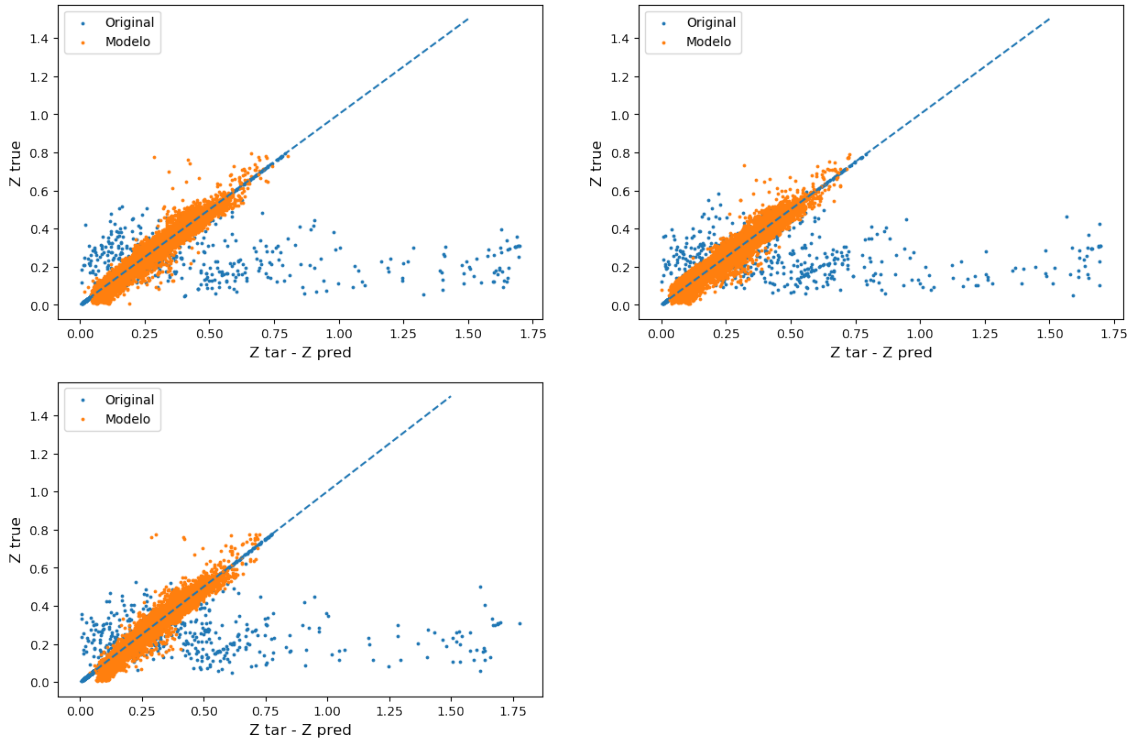


Figure 4.3: Results of each model on the development set.

The gathering capacity of the model to move the blue dots towards the TRUEZ = TARZ line present on the training set is also visible on the evaluation set, were the r^2 scores reaches 93% only.

4.2.2 Kernel Ridge Regression (KRR) Model

The grid-search and cross-validation on the train-development set use the r^2 measure to select the best parameters.

```
--------------------------------------------------------------------------------
Model 1:
Best parameters set found on development set:

{'alpha': 0.0001, 'gamma': 0.2, 'kernel': 'rbf'}

Grid scores on development set for r2:

0.986 (+/-0.002) for {'alpha': 0.001, 'gamma': 0.1, 'kernel': 'rbf'}
0.988 (+/-0.002) for {'alpha': 0.001, 'gamma': 0.2, 'kernel': 'rbf'}
0.987 (+/-0.002) for {'alpha': 0.0001, 'gamma': 0.1, 'kernel': 'rbf'}
```

```
0.989 (+/-0.002) for {'alpha': 0.0001, 'gamma': 0.2, 'kernel': 'rbf'}

r2 score computed on the full evaluation set:

0.990
-------------------------------------------------------------------------
Model 2:
Best parameters set found on development set:

{'alpha': 0.001, 'gamma': 0.2, 'kernel': 'rbf'}

Grid scores on development set for r2:

0.986 (+/-0.001) for {'alpha': 0.001, 'gamma': 0.1, 'kernel': 'rbf'}
0.988 (+/-0.002) for {'alpha': 0.001, 'gamma': 0.2, 'kernel': 'rbf'}
0.987 (+/-0.002) for {'alpha': 0.0001, 'gamma': 0.1, 'kernel': 'rbf'}
0.987 (+/-0.004) for {'alpha': 0.0001, 'gamma': 0.2, 'kernel': 'rbf'}

r2 score computed on the full evaluation set:

0.988
-------------------------------------------------------------------------
Model 3:
Best parameters set found on development set:

{'alpha': 0.001, 'gamma': 0.2, 'kernel': 'rbf'}

Grid scores on development set r2:

0.987 (+/-0.001) for {'alpha': 0.001, 'gamma': 0.1, 'kernel': 'rbf'}
0.988 (+/-0.002) for {'alpha': 0.001, 'gamma': 0.2, 'kernel': 'rbf'}
0.987 (+/-0.003) for {'alpha': 0.0001, 'gamma': 0.1, 'kernel': 'rbf'}
0.987 (+/-0.003) for {'alpha': 0.0001, 'gamma': 0.2, 'kernel': 'rbf'}

r2 score computed on the full evaluation set:

0.9899
```

The results on training set of the best models is shown in Figure 4.4. On training, the model shows better results than SVR by gathering the blue dots (original data)over the orange dots (the model results after applying it to the training data) on a tighter region. The blue scatter is substantially diminished and the data is clustered along de TRUEZ =

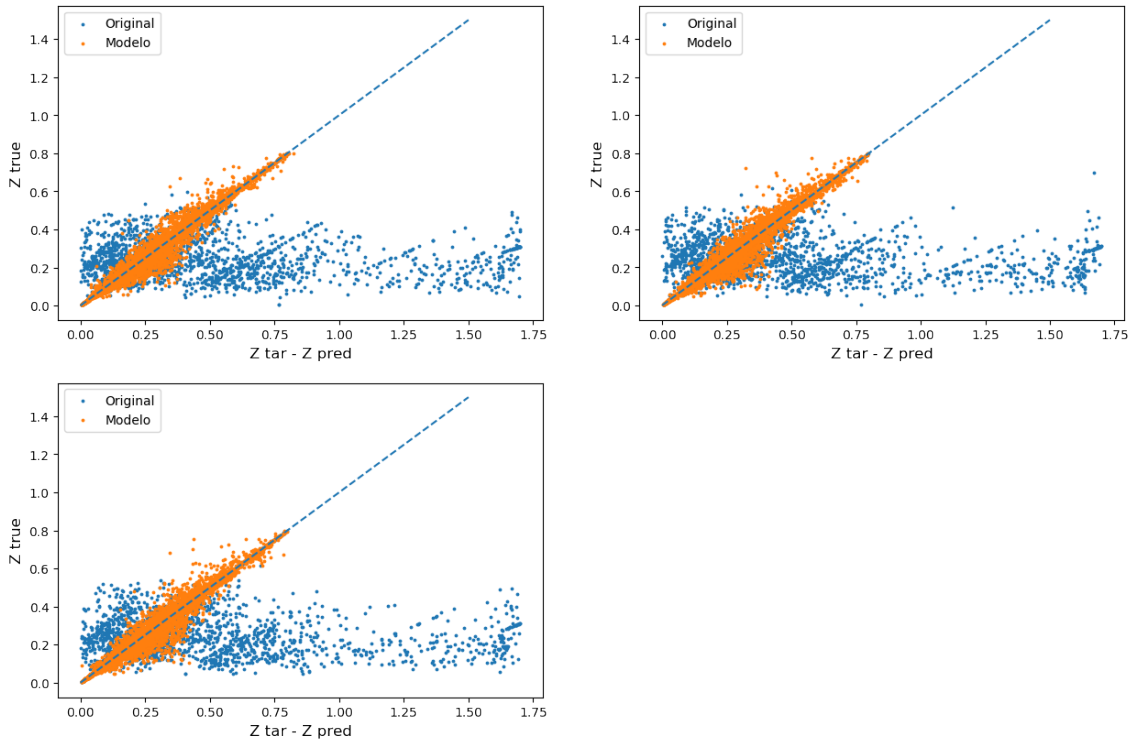


Figure 4.4: Results onthe training set after training the KRR model.

TARZ line. The result on the evaluation set for each model is show in Figure 4.5. Figure

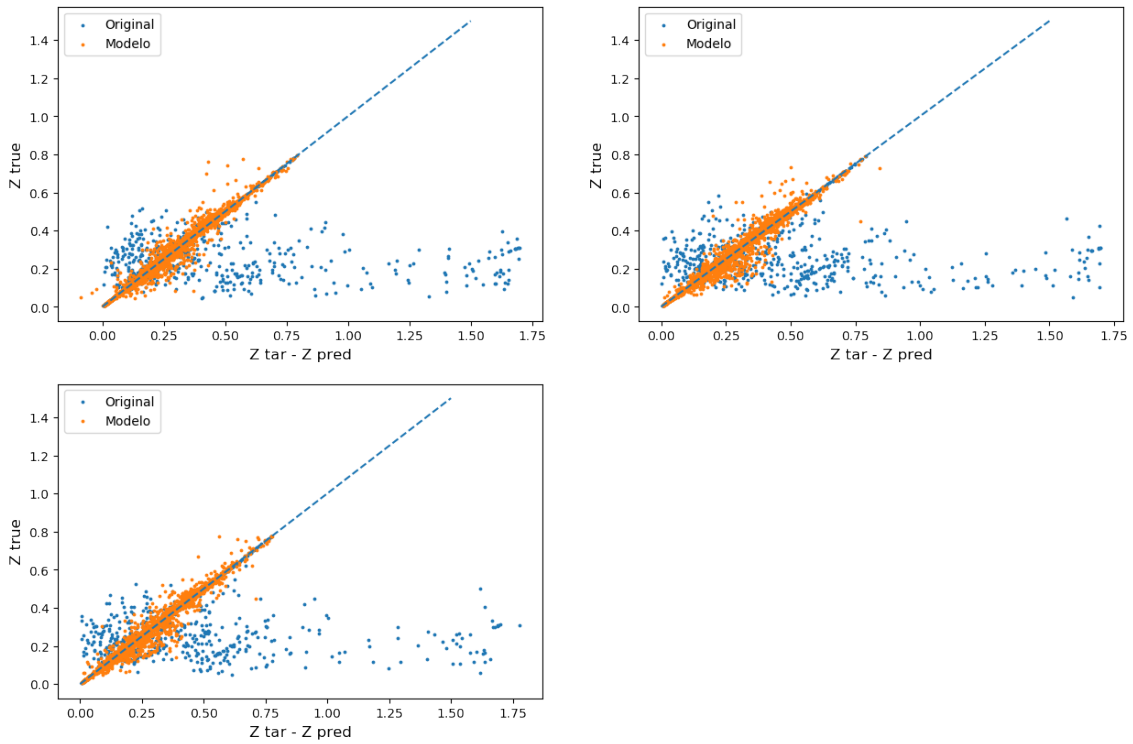


Figure 4.5: Results of each model on the development set.

4.5 shows how the model generalize to unseen data. The gathering capacity of the model to move the blue dots towards the TRUEZ = TARZ line present on the training set is also visible on the evaluation set, were the r^2 scores reaches 99%, a more better result than

that obtaining using SVR.

4.2.3 Model Ensemble

The results from the KRR are much better than those of the SVR, also noting that the KRR model takes about half the time to train. Given that we have three different models to predict on the same dataset, we have to make this model to predict a single output. For this, we tried taking the maximum, the average or a weighted average of the model.

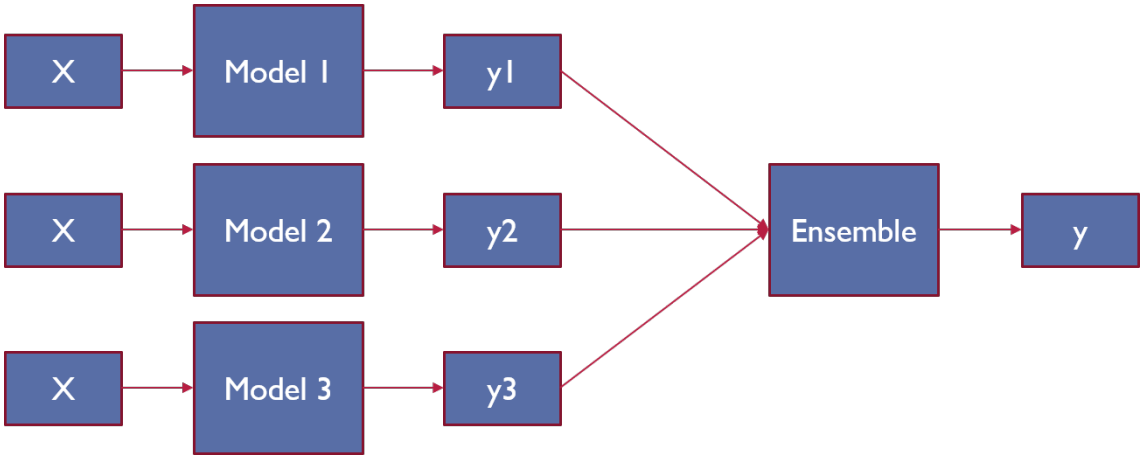


Figure 4.6: Graphic description of the ensemble model. It is formed based on the trained KRR models 1, 2 and 3.

Figure 4.6 shows this in a diagram, it is important to note that the ensemble is made over the **trained** models, that means that the parameters of the models are not re-learned in the ensemble. The only ensemble model that involves training, is the weighted average, where the weights are select by training a linear model. The results of the models on the **test** subset (unseen 25%) are shown in Table 4.1.

Model	r^2
Model 1	0.958
Model 2	0.956
Model 3	0.962
Model avg	0.962
Model max	0.949
Model w	0.983

Table 4.1: Results on the **test** set

Table 4.1 shows that the weighted average of the three models is the best model in terms of accuracy, taking into account that this are the predictions of the models on the totally-unseen test data corresponding to the 25% of the original BGS dataset.

4.3 Selección de modelo

The class of models evaluated is that of kernels models, the complexity of both, the SVR and the KRR is very similar, however the KRR gave better results. This could be atributed to the lack of tuning of the ϵ parameter in SVR, due to the time of computation of SVR training. Smaller values of ϵ, may result in a tighter region over the TRUEZ = TARZ line, therefore increasing the r^2. The weighted average model has at least 3 times more parameters that any model individually, its results are much better. Therefore, the best model, that keeps simplicity and very good results is the weighted average model.

4.4 Conclusions

We evaluated two types of kernels method, the support vector regression (SVR) and the kernell ridge regression (KRR). The dataset was split in a training part (75%) and a test part (25%). Each model was trained tree times on subset of the training part of size 100.000, this subset were also subdivide in a 80-20 development - evaluation datasets for the application of grid search and cross validation. The KRR gave better results than the SVR, also, the three KRR models were ensemble as a weighted average, where the weights were found using a linear regressor. The ensemble average KRR model reached a r^2 of 0.98 on the test set.

Chapter 5

Conclusions

It was found that for the BGS dataset that the variables FLUX_G, FLUX_R, FLUX_Z, FLUX_W1, FLUX_W2 and Z (target Z), can be used effectively to approximate the instrument measurements to the ground true using kernel methods. Using a ensemble model consisting of three KRR (Kernel Ridge Regression) trained on subsamples of size 100.000, we were capable of aproximate the DESI instrument redshift measurements to its true value up to a $r^2 = 0.98$. This model shows great potential to further enhancement, i.e. the fine tunning of the model hyper-parameters, different preprocessing methods, or even different classes of machine learning algorithms.

It is important to execute a training time evaluation as a function of the cluster resources and the parallelization algorithms, in order to know from the beginning the expected times of the algorithms and the correct selection of the sub-dataset size.

Although the trained models didn´t work for other clases of galaxies, probably because the different distributions of fluxes, we infer that the same methodology can be used starting by training the models in the corresponding datasets. Moreover, a single model can be used for all the classes of targets by adding the categorical features of TYPE and SUBTYPE. We expected that by doing so, the results of this work can be replicated and applied to all the targets in the DESI measurements.