Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **449,** 1275–1283 (2015) doi:10.1093/mnras/stv373

Feature importance for machine learning redshifts applied to SDSS galaxies

Ben Hoyle,[1,2]★ Markus Michael Rau,[1] Roman Zitlau,[1] Stella Seitz[1,3] and Jochen Weller[1,2,3]

[1] *Universitaets-Sternwarte, Fakultaet fuer Physik, Ludwig-Maximilians Universitaet Muenchen, Scheinerstr. 1, D-81679 Muenchen, Germany*
[2] *Excellence Cluster Universe, Boltzmannstr. 2, D-85748 Garching, Germany*
[3] *Max Planck Institute for Extraterrestrial Physics, Giessenbachstr. 1, D-85748 Garching, Germany*

Accepted 2015 February 3. Received 2015 January 26; in original form 2014 October 20

ABSTRACT

We present an analysis of importance feature selection applied to photometric redshift estimation using the machine learning architecture Decision Trees with the ensemble learning routine ADABOOST (hereafter RDF). We select a list of 85 easily measured (or derived) photometric quantities (or 'features') and spectroscopic redshifts for almost two million galaxies from the Sloan Digital Sky Survey Data Release 10. After identifying which features have the most predictive power, we use standard artificial Neural Networks (aNNs) to show that the addition of these features, in combination with the standard magnitudes and colours, improves the machine learning redshift estimate by 18 per cent and decreases the catastrophic outlier rate by 32 per cent. We further compare the redshift estimate using RDF with those from two different aNNs, and with photometric redshifts available from the Sloan Digital Sky Survey (SDSS). We find that the RDF requires orders of magnitude less computation time than the aNNs to obtain a machine learning redshift while reducing both the catastrophic outlier rate by up to 43 per cent, and the redshift error by up to 25 per cent. When compared to the SDSS photometric redshifts, the RDF machine learning redshifts both decreases the standard deviation of residuals scaled by $1/(1+z)$ by 36 per cent from 0.066 to 0.041, and decreases the fraction of catastrophic outliers by 57 per cent from 2.32 to 0.99 per cent.

Key words: catalogues – surveys – galaxies: distances and redshifts.

1 INTRODUCTION

Large-scale photometric galaxy surveys require precise redshift information to maximize information about cosmological parameters. Obtaining accurate spectroscopic redshifts is costly and time intensive and is typically only performed for a small subsample of all galaxies. Conversely, the measurement of multiband photometric properties of galaxies is much cheaper. The compromise is then to attempt to extract less accurate redshift information from photometrically measured properties, but applied to the full galaxy sample. This paper examines which photometric properties, or 'features', of the galaxies are best suited to this task by using feature importance analysis within standard machine learning architectures.

Photometric redshifts are often derived from galaxy spectral energy distribution (hereafter SED) templates. The template redshift approach is well studied and is physically motivated. We know how the measured flux of a fiducial galaxy will change with redshift, and we can employ our knowledge of stellar populations and their evolution to predict how the SED and therefore fluxes and colours of galaxies will change as a function of redshift and galaxy type. However, the encoding of the complex stellar physics, the computational time required to generate the templates, coupled with our uncertainty in the stellar population models and observed measurement error, combine to produce redshift estimates which are little better than non-parametric techniques (see e.g. Hildebrandt et al. 2010; Dahlen 2013, for an overview of different techniques).

Machine learning methods offer a non-parametric alternative to template methods. Generally the 'machine architecture' learns how to combine and (or) weight and (or) cluster the photometric galaxy properties to produce a machine learning redshift. The machine then examines the best combinations (and) or clustering (and) or weighting to minimize the difference between the spectroscopic redshift (of a training sample) and the machine learning redshift.

A major advantage of the template method over the machine learning method is the need of the latter to have a well-defined training set which spans the input feature parameter space of interest. On the other hand, one may view the need to generate templates and, importantly, a non-biased sample of templates, an equally large obstacle in order to produce reasonable template method redshifts.

★ E-mail: benhoyle1212@gmail.com

There has also been work to combine template and machine learning techniques (e.g. Feldmann et al. 2006; Ilbert et al. 2009) so the line of distinct has become less clear.

Non-parametric photometric redshift estimation techniques have been developing since Connolly et al. (1995) and moved into the field of machine learning with the popular artificial Neural Network (aNN) code called ANNZ (Collister & Lahav 2004). Since then a plethora of machine learning architectures, including Random Decision Trees, have been applied to the problem of redshift estimation (see e.g. Sánchez et al. 2014, for a list of routine comparisons), and to estimate the full redshift probability distribution function (Gerdes et al. 2010; Carrasco Kind & Brunner 2013). Machine learning architectures have also had success in other fields of astronomy such as galaxy morphology identification, and star & quasar separation (see for example Lahav 1997; Yeche et al. 2009).

Most recently Brescia et al. (2014b) applied an advanced type of deep aNN to a subset of galaxies drawn from the Sloan Digital Sky Survey (hereafter SDSS; York & SDSS Collaboration 2000) Data Release 9 (Ahn et al. 2012). The galaxy training sample was selected to be clean of artefacts, and to be confirmed spectroscopically as a galaxy. The resulting machine learning redshift error for this clean subset of galaxies is $\sigma_z = 0.023$, several factors smaller than the photometric redshift available in the SDSS CasJobs (Nolan & Ani 2008) interface for the same sample of galaxies.

The machine learning community uses the nomenclature 'features' for items which are input into a machine architecture. For our purposes, the features can be any easily measured, or derived, photometric quantities that are available for each galaxy. For example standard machine learning redshift analysis uses a set of input features drawn from either fluxes or magnitudes, or a pivot magnitude and colours.

However, one has the freedom to choose other features which are easily measured photometrically and which may also scale with distance. For example it is conceivable that the observed galaxy size, or the isophotal radius in some (or all) bands may also encode redshift information. One may also find that galaxy inclination, or galaxy type as measured by the SDSS 'fracDev' parameter, or Stokes parameters, may also be valuable to identify (perhaps morphological) subsamples which may each have different redshift scalings (see Yip et al. 2011). Given the quantity of easily obtained photometric parameters, and their ease of accessibility, it seems pertinent to explore if the addition of extra features can indeed improve machine learning redshift estimates.

Indeed early work using aNNs by Tagliaferri et al. (2003) find that the inclusion of radii and fluxes in addition to magnitudes improves the machine learning redshift estimation compared with using just magnitudes. However, Singal et al. (2011) use more derived morphological input features and a principle component type analysis to show that the addition of the examined features do not drastically improve machine learning redshift estimation. Furthermore Li et al. (2006) and Brescia et al. (2013) explore different magnitude definitions as input features of aNNs, and find that some magnitudes produce more accurate machine learning redshift estimates than others (e.g. dereddened model magnitudes from the SDSS).

We expand these previous result by compiling 85 standard (e.g. magnitudes) and extended photometric features, and then use feature importance to determine which features have the most predictive power when estimating the galaxy redshift. This is performed using standard feature selection analysis well known to the machine learning community. A full feature importance analysis with so many different features has yet to be applied to machine learning redshift estimation.

We examine the use of the following machine learning architectures: Decision Trees combined using the ADABOOST algorithm to perform the feature importance and to measure redshifts, and standard aNNs to see how the effect of selecting different features changes the recovered machine learning redshifts. We note that some of the aforementioned machine learning architectures are extremely scalable, and can be performed in a matter of tens of seconds on today's desktop computers with sample sizes of millions.

The format of the paper follows. We describe the data sample and the list of measured and derived photometric features in Section 2. We continue by detailing the machine learning architectures applied in this work, and introduce the feature importance in Section 3. We describe the analysis and present results in Section 4, and conclude and discuss in Section 5.

2 DATA

The data in this study are drawn from SDSS Data Release 10 (DR10; Ahn et al. 2014). The SDSS I-III uses a 4-meter telescope at Apache Point Observatory in New Mexico and has CCD wide field photometry in five bands (u, g, r, i, z; Smith et al. 2002; Gunn et al. 2006), and an expansive spectroscopic follow up program (Eisenstein 2011) covering π radians of the northern sky. The SDSS collaboration has obtained approximately 2 million galaxy spectra using dual fibre-fed spectrographs. An automated photometric pipeline performed object classification to a magnitude of $r \approx 22$ and measured photometric properties of more than 100 million galaxies. The complete data sample, and many derived catalogues such as the photometric redshift estimates, are publicly available through the CasJobs server.[1]

The SDSS is well suited to the analysis presented in this paper due to the enormous number of photometrically selected galaxies with spectroscopic redshifts to use as training, cross-validation and test samples. We select 1958 727 galaxies from CasJobs with both spectroscopic redshifts and photometric properties. In detail we run the following MYSQL query in the DR10 schema:

```
SELECT s.specObjID, s.objid, s.ra,s.dec,
p.z AS photoz, p.zerr AS photoz_err,
s.z AS specz, s.zerr AS specz_err,
s.dered_u,s.dered_g,s.dered_r,s.dered_i,
s.dered_z,s.modelMagErr_u,s.modelMagErr_g,
s.modelMagErr_r,s.modelMagErr_i,s.modelMagErr_z,
s.type as specType, q.type as photpType,
q.petroRad_u,q.petroRad_g,q.petroRad_r,
q.petroRad_i,q.petroRad_z,
q.petroRadErr_u,q.petroRadErr_g,q.petroRadErr_r,
q.petroRadErr_i,q.petroRadErr_z,
q.deVRad_u,q.deVRadErr_u,q.deVRad_g,q.deVRadErr_g,
q.deVRad_r,q.deVRadErr_r,
q.deVRad_i,q.deVRadErr_i,q.deVRad_z,q.deVRadErr_z,
q.extinction_u,q.extinction_g,q.extinction_r,
q.extinction_i,q.extinction_z,
q.psfMag_u,q.psfMagErr_u,
q.psfMag_g,q.psfMagErr_g,
q.psfMag_r,q.psfMagErr_r,
q.psfMag_i,q.psfMagErr_i,
q.psfMag_z,q.psfMagErr_z,
q.fiberMag_u,q.fiberMagErr_u,
```

[1] skyserver.sdss3.org/CasJobs

```
q.fiberMag_g,q.fiberMagErr_g,
q.fiberMag_r,q.fiberMagErr_r,
q.fiberMag_i,q.fiberMagErr_i,
q.fiberMag_z,q.fiberMagErr_z,
q.expAB_u,q.expRad_u,q.expPhi_u,
q.expAB_g,q.expRad_g,q.expPhi_g,
q.expAB_r,q.expRad_r,q.expPhi_r,
q.expAB_i,q.expRad_i,q.expPhi_i,
q.expAB_z,q.expRad_z,q.expPhi_z

INTO mydb.specPhotoDR10v2 FROM SpecPhotoAll AS s

JOIN photoObjAll AS q
ON s.objid=q.objid AND q.dered_u>0
AND q.dered_g>0 AND q.dered_r>0
AND q.dered_z>0 AND q.dered_i>0
AND q.expAB_r >0

LEFT OUTER JOIN Photoz AS p ON s.objid=p.objid
```

We apply the SDSS extinction corrections to the point spread function (hereafter psf) and fibre magnitudes, and further only select galaxies that have a photometric galaxy classification *type* = 3, have spectroscopic redshifts, *r*-band magnitudes, and radii greater than zero. This reduces the sample size to 1922 231 galaxies.

2.1 SDSS DR10 photometric redshifts

The SDSS photometric redshifts are generated using a hybrid technique of the template method (Budavári et al. 2000) and a machine learning component using *k*-nearest neighbours (Csabai et al. 2007) technique as described in Abazajian et al. (2009). We hereafter refer to this combined method as 'template-ml'. The SDSS template-ml photometric redshifts are available from within CasJob by using the above SQL query.

2.2 Input features

Table 1 shows the list of photometric features used in this work. This is a large but non-exhaustive list of possible input features. There are still more photometric features one may choose to use, such as Petrosian magnitudes, other apertures, or more detailed surface profiles (see e.g. Singal et al. 2011).

These photometric features are drawn from the following categories. Magnitudes: corresponding to magnitudes measured in different bands and apertures, and colour combinations created from them; Radii: measurements of sizes in different bands and with differing definitions; Morphology: how much of the light profile is best fitted by one profile compared to others; and Shapes: ratio of major and minor ellipses measured in different bands and the Means Stokes parameters in each band. We list the full set of input features in Table 1 and note that their full description can be found on the Sky Server web page.[2]

For each feature dimension, we perform feature rescaling by subtracting the mean of the feature distribution and dividing by two times the standard deviation. Feature rescaling allows features with potentially vastly different scales to be given equal weight in the analysis. Throughout the remaining paper all references to features refer to these rescaled features. The aim of this work is to identify which of these features provides the most predictive power for redshift estimation. In order to select these features, we perform feature importance, described in Section 3.4. We then obtain a ranking of features, in which the features which have the highest rank have the most predictive power and are selected as inputs into the final model.

Table 1. The complete list of the input photometric features used in this work. The Means Stokes parameters are shape features. A full description of each of the parameters can be found on the SDSS Sky Server web page.

Description	Feature name
Magnitudes	dered_u dered_g dered_r dered_i dered_z psfMag_u psfMag_g psfMag_r psfMag_i psfMag_z fiberMag_u fiberMag_g fiberMag_r fiberMag_i fiberMag_z
Radii	petroRad_u petroRad_g petroRad_r petroRad_i petroRad_z expRad_u expRad_g expRad_r expRad_i expRad_z deVRad_u deVRad_g deVRad_r deVRad_i deVRad_z
Colours	dered_z-dered_i dered_z-dered_r dered_z-dered_g dered_z-dered_u dered_i-dered_r dered_i-dered_g dered_i-dered_u dered_r-dered_g dered_r-dered_u dered_g-dered_u fiberMag_z-fiberMag_i fiberMag_z-fiberMag_r fiberMag_z-fiberMag_g fiberMag_z-fiberMag_u fiberMag_i-fiberMag_r fiberMag_i-fiberMag_g fiberMag_i-fiberMag_u fiberMag_r-fiberMag_g fiberMag_r-fiberMag_u fiberMag_g-fiberMag_u psfMag_z-psfMag_i psfMag_z-psfMag_r psfMag_z-psfMag_g psfMag_z-psfMag_u psfMag_i-psfMag_r psfMag_i-psfMag_g psfMag_i-psfMag_u psfMag_r-psfMag_g psfMag_r-psfMag_u psfMag_g-psfMag_u
Profile	fracDeV_u fracDeV_g fracDeV_r fracDeV_i fracDeV_z
Ellipticity	expAB_u expAB_g expAB_r expAB_i expAB_z deVAB_u deVAB_g deVAB_r deVAB_i deVAB_z
Means Stokes	q_u u_u q_g u_g q_r u_r q_i u_i q_z u_z

2.3 Training, cross-validation and test subsets

We follow traditional machine learning nomenclature and methodology by randomly sub-dividing the galaxy catalogue into a training, cross-validation and test samples with proportions 50, 25, 25 per cent, respectively. The training sample is used to train the machine learning system for a given architecture and hyperparameter set. One uses the cross-validation sample to select the best values for the hyperparameters of the learned system. Once the set of hyperparameters has been decided upon, neither the training nor cross-validation sample provide a bias free estimate of the true error. In these cases, the test sample is used to measure the true ability of the learned machine to generalize to a new data set.

[2] skyserver.sdss3.org/public/en/help/browser/browser.aspx

3 MACHINE LEARNING METHODS

Below we describe the two aNN programs used in this work and the machine learning frameworks Decision Trees with the ensemble learning code ADABOOST.

3.1 Artificial Neural Networks

We use two different aNN architectures. The first is the Java based applet Photoraptor (Brescia & Cavuoti 2014a; Cavuoti, Brescia & Longo 2014) and the second is the FORTRAN code FANN (Nissen 2003) which is callable from the command line and has wrappers in many languages including PYTHON.

Photoraptor is a standard multilayer perceptron with up to two (feed forward) hidden layers which trains the connections between neurons efficiently using a quasi-Newton algorithm. We follow Brescia et al. (2014b) and use an architecture of two hidden layers of size (11,4).

FANN extends standard aNN architecture by incrementally building hidden layers and determining connections using the Cascade2 training algorithm (Fahlman & Lebiere 1990). Cascade2 training starts with an empty set of hidden layers and incrementally trains and adds one (multiply connected) neuron until either a user-specified maximum number have been added, or until the training error reaches some user-specified threshold.

The hyperparameters of the aNNs explored in this work are the number of training examples (the training sample size), and the number of input features per training example, the number of hidden layers and neurons, and the method to learn the best connections between neurons.

3.2 Decision Trees

We use the PYTHON package scikit-learn (Pedregosa et al. 2011) and the included implementation of Decision Trees for regression (Breiman et al. 1984). There exist other public implementations of trees and forests for Classification and Regression, some of which estimate the full shape of the machine learning redshift probability distribution function (e.g. Carrasco Kind & Brunner 2013).

The Decision Tree Regressor sub-divides (or branches) the N data with respect to the feature space $\boldsymbol{f}_i$ into τ branches $\mathcal{B}_\tau$ which end in l leaf nodes per tree. The branches are constructed such that the spectroscopic redshift $z_{\mathrm{spec},i}$ in each leaf has a low mean squared error

$$\mathrm{MSE} = \frac{1}{N}\sum_{\tau=1}^{l}\sum_{\boldsymbol{f}_i\in\mathcal{B}_\tau}\left(z_{\mathrm{spec},i} - \left\langle z_{\mathrm{spec},\tau}\right\rangle\right)^2 , \quad (1)$$

with respect to the mean spectroscopic redshift

$$\left\langle z_{\mathrm{spec},\tau}\right\rangle = \frac{1}{N_\tau}\sum_{\boldsymbol{f}_i\in\mathcal{B}_\tau} z_{\mathrm{spec},i} , \quad (2)$$

Where N_τ is the number of objects in each leaf. The branches of the tree correspond to regions of input feature space. The machine learning redshift of a new object is obtained by assigning the value of equation (2) to the final leaf that the data falls upon. The tree is grown recursively such that the mean spectroscopic redshift in each leaf is similar to the spectroscopic redshift of the objects in the leaf. We measure this similarity using the mean squared error function (equation 1). The selection of the best input feature to split on, and the best splitting point, is determined using an exhaustive search to minimize equation (1). The tree is grown until each leaf node contains N_τ objects, where N_τ is a hyperparameter of the model. We then determine the feature importance score by summing (and normalizing) the decrease in equation (1) for each feature.

Computing a single Decision Tree is incredibly fast. This implementation can partition one million galaxies along 85 feature dimensions in a few tens of seconds using only one CPU core. Predicting a redshift for new data is also very fast, requiring just seconds on a data set of size half of a million.

3.3 ADABOOST

The power of the Decision Trees can be enhanced by combining many trees together to make a forest. We select randomly from input data to produce each forest so use the terminology RDF throughout the paper. One method to achieve this is by using an algorithm called Adaptive Boosting or ADABOOST (Drucker 1997; Freund & Schapire 1997). In this work, we define the collection of trees constructed using ADABOOST as a forest. This should not be confused with normal random forests in machine learning, which build trees simultaneously. We give a brief overview of the algorithm below, and refer the reader to Drucker (1997) for a more detailed description of the algorithm used in the scikit-learn routine. We note that Gerdes et al. (2010) also use ADABOOST and trees, but they determine the full shape of the machine redshift probability distribution function using SDSS magnitudes as input features.

The basic idea of boosting is to improve the performance of a base learner, in our case the Decision Tree Regressor, by using multiple models which put more weight on elements in the training set which have large prediction errors.

The algorithm described in Drucker (1997) first assigns equal weight $w_i = 1$ to each galaxy in the training set. Subsequently, one trains a Decision Tree Regressor on a new training set of size N, by bootstrap selecting N samples with replacement from the original training set. Each element has a probability to be selected given by

$$p_i = \frac{w_i}{\sum_{i=1}^{N} w_i} . \quad (3)$$

This produces a new model which is added to the ensemble of models. The training set loss L_i, for each element i is calculated as

$$L_i = \frac{|z_{\mathrm{phot}}(\boldsymbol{f}_i) - z_{\mathrm{spec},i}|}{\sup_j |z_{\mathrm{phot}}(\boldsymbol{f}_j) - z_{\mathrm{spec},j}|} , \quad (4)$$

where $z_{\mathrm{phot}}(\boldsymbol{f}_j)$ is the function represented by the corresponding tree. Note that L_i is normalized in such a way, that $L_i \in [0, 1]$. We can then calculate the average loss $\overline{L}$ of the model using

$$\overline{L} = \sum_{i=1}^{N} L_i\, p_i , \quad (5)$$

where the sum runs over all elements in the training set. The confidence β for a specific model is defined by

$$\beta = \frac{\overline{L}}{1-\overline{L}} , \quad (6)$$

and the weights for each model are iteratively updated by multiplying the weights for each element in the training set by β^{1-L_i}.

The weight update procedure gives less weight to elements with a low-prediction error L_i and therefore these objects are less likely to be included in the training set drawn in the next boosting iteration. This focuses subsequent learners on elements with a high prediction error (Drucker 1997; Freund & Schapire 1997). We train a number of Decision Tree Regressors in this way and update the weights for

the training set. The number of trees M included into the ensemble is a hyperparameter of the model. If we query a new object with input features $\boldsymbol{f}_i$, we obtain a prediction $z_{\mathrm{phot},j}(\boldsymbol{f}_i)$ for each tree in the ensemble $j \in 1, \ldots, M$. The final machine learning redshift prediction $z_{\mathrm{phot}}(\boldsymbol{f}_i)$ is then given as the weighted median of the redshift predictions of the models in the ensemble with respect to $\log(1/\beta_j)$ (as described in Drucker 1997).

3.4 Feature importance selection

One noteworthy feature of RDF is the ability to determine which of the feature dimensions encode the most information about the quantity of interest, which here is the machine learning redshift. The scikit-learn implementation of Decision Trees uses the Gini importance (described below) to determine the predictive power of each feature. In detail, the ADABOOST routine combines the feature importances for each tree to create a forest-wide feature importance following this procedure. First, the feature importance of each Decision Tree is determined and then the same weight is applied to each feature importance value as applied to the tree when constructing the forest. The final output value of the feature importance provided by ADABOOST is the sum of the individual tree importances normalized by the sum of the weights applied to each tree.

The Gini importance is constructed from the Gini coefficient, which is the value of the Mean Squared Error (MSE) (equation 1) of the items on each branch. The larger the MSE the larger the Gini coefficient. As more sub-branches are formed the data on each sub-branch become more homogeneous which reduces the Gini coefficient. The Gini importance measures the reduction in the Gini coefficient from the parent branch to the child sub-branches. For our purposes, the higher the Gini importance, the more the feature is able to separate the training data into similar redshift groups, and therefore the more predictive power the feature has.

In summary, the more branches in the Decision Tree that a particular feature dimension has, the more predictive power it has when estimating the redshift compared with other feature dimensions.

4 ANALYSIS AND RESULTS

We first determine which are the most important features using RDFs. We then document how the standard aNN machine learning redshift is improved by the addition of the most important features. We then present what effect the size of the training sample has on the recovered machine learning redshift and finally show the further improvement in the photometric redshift of SDSS galaxies when using RDFs.

4.1 Feature importance

To determine feature importance we construct 25 forests, and for each forest we vary the following hyperparameters: the number of trees, the number of objects on each leaf node, the random seed and the size of the training set. Upon completion of each forest ADABOOST returns the list of input features with their representative importance weights. We rank the features by weight and extract the top 1, 2, 3 features for each forest. We construct a results cube which lists how often each feature has made it to a given importance rank.

In Fig. 1, we show the relative importance of different photometric features when determining a machine learning redshift. The first column shows the most important, or top ranked, feature after each iteration. The second column shows the second most important feature. The height of the colour bar is the occurrence rate that the feature has been deemed to be the most important. Note that for each forest we randomly select the machine learning architecture hyperparameters and the training sample size. The x-axis labels corresponds to the ranked importance of the features and we label each feature on the figure. The colours represent the same feature across each column.

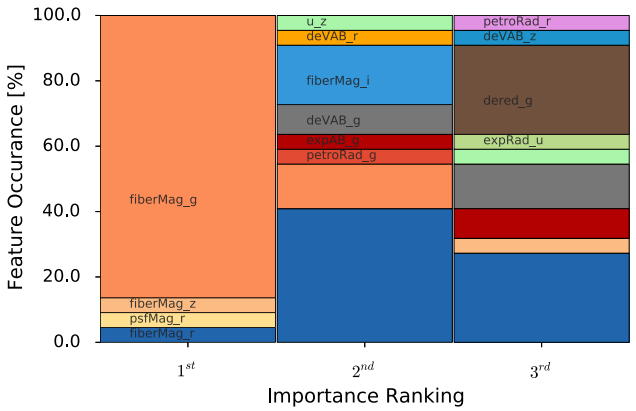

Figure 1. The relative importance of different photometric features when determining a machine learning redshift using SDSS data. The first column shows the most important, or top ranked, feature after each iteration. The second column shows the second most important feature. The height of the colour bar is the occurrence rate that the feature has been deemed to be the most important. Note that for each forest we randomly select the machine learning architecture hyperparameters and training sample size. The x-axis labels corresponds to the ranked importance of the features and we label each feature. The colours represent the same feature across each column.

Surprisingly, we find that the most important feature is the value of the fibre magnitude measured in the g band, which, to the authors knowledge, has never been used in machine learning frameworks. We note that the SDSS fibre magnitude is a measure of the flux within a small (2 arcsec) radius around the galaxy centre and is measured for all galaxies, not only the spectroscopic subset. We also note that the fibermag_g is not only the most important feature, but it is often more important by a factor of 3 more than the second most importance feature.

Given that we are using spectroscopic galaxies and selecting only galaxies with clean spectra, this implies that only galaxies whose flux is well defined within the fibre radius make the finally selection. Therefore, these galaxies fibre apertures are probably a reasonable approximation of the underlying galaxy SED, and thus the apparent magnitude will scale with distance.

To improve statistical significance we generate a further 350 further forests, and repeat the feature analysis. We find fiberMag_g is still the most important, top 1, feature 67 per cent of the time. However, 25 other features also appear at least once in the top 1 feature ranking, and we decide to not include these in Fig. 1 to improve readability.

As an illustrative example we examine how the top features scale as a function of redshift and show this illustrative plot in Fig. 2. Recall that each of the input features are pre-scaled by subtracting the mean and dividing by twice the standard deviation, see Section 2.2. We have arbitrarily rescaled the features, in a manner such as some of the machine architectures may use. Note that this is an illustrative explanation of the power of the most important features at determining galaxy redshift. The top panel shows two of the standard features used in machine learning (see e.g. Brescia et al. 2014b). The bottom panel uses the top three most important features as

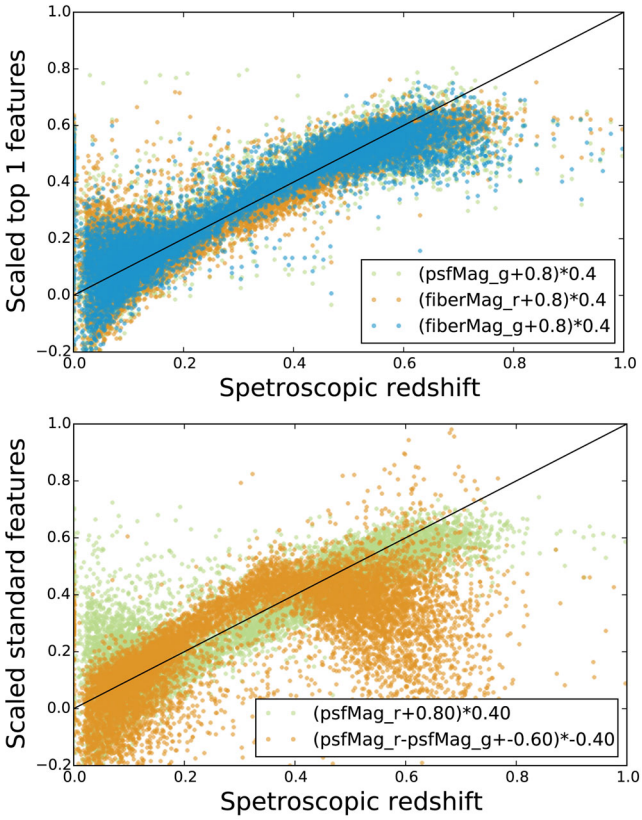

Figure 2. An illustrative example of the power of the most important features at determining galaxy redshift. The top panel shows two standard features used in machine learning (Brescia et al. 2014b). The bottom panel shows the three most important features as determined by this work. The x-axis is the true redshift and the y-axis shows the arbitrarily scaled features. The same sample of galaxies is shown in both plots. Note that all features are pre-scaled by subtracting the mean and dividing by twice the standard deviation, see Section 2.2.

determined by this work. The data in this illustrative example is drawn from the test sample, which has not been used during machine training, and is therefore independent of the analysis. The same sample of galaxies is shown in both plots.

We also remove the fibre magnitudes and fibre colours completely from the analysis, and repeat the feature importance as above. We find that the dered_g model magnitude is now the most important feature 70 per cent of the time.

4.2 The effect of the extended features

In this section, we show what effect the use of extended features has on the machine learning redshift error estimated using popular frameworks. We choose two types of aNN, and fix their architectures in one approach, and allow it to vary in the other.

In the first approach we use Photoraptor and fix the size of the training and test sets to be 120139, 360417, and the number of hidden layers to two and hidden units to (11,4). We note that here, the sum of the training and cross-validation data set corresponds to 25 per cent of the full data sample. Following Brescia et al. (2014b), we refer to the following features as 'standard': psfMag_g-psfMag_u,psfMag_r-psfMag_g,psfMag_i-psfMag_r,psfMag_z-psfMag_i,psfMag_r. We use the results from Section 4.1 to determine which are the top 1 and 2, and then top 1, 2 and 3, most important 'extended' features. The extended features are labelled in Fig. 1. We note that the top 1 and 2 features have dimension 10, and the top 1, 2 and 3 have dimension 14.

Table 2. The values of the mean μ, standard deviation σ and the percentage of outliers for the residuals Δ_z between the true redshift and the Photoraptor machine learning redshift. We also show the results for the residuals scaled by $1/(1+z)$. We show the measured values for different input features sets (see text) and fix the training sample size and machine architecture hyperparameters. Standard features are the psf magnitudes and colours, and the top 1, 2, 3 features are those labelled in Fig. 1.

Input features	Measurement	Value
Standard	$\mu_{\Delta_z} \pm \sigma_{\Delta_z}$	-0.0001 ± 0.075
	$\mu_{\Delta_z/(1+z)} \pm \sigma_{\Delta_z/(1+z)}$	-0.003 ± 0.055
	$\lvert\Delta_z/(1+z)\rvert > 0.15$	1.74 per cent
Top1&2	$\mu_{\Delta_z} \pm \sigma_{\Delta_z}$	0.0001 ± 0.068
	$\mu_{\Delta_z/(1+z)} \pm \sigma_{\Delta_z/(1+z)}$	-0.002 ± 0.049
	$\lvert\Delta_z/(1+z)\rvert > 0.15$	1.34 per cent
Standard&top1&2	$\mu_{\Delta_z} \pm \sigma_{\Delta_z}$	-0.0001 ± 0.066
	$\mu_{\Delta_z/(1+z)} \pm \sigma_{\Delta_z/(1+z)}$	-0.002 ± 0.046
	$\lvert\Delta_z/(1+z)\rvert > 0.15$	1.22 per cent
Standard&top1&2&3	$\mu \pm \sigma$	0.0001 ± 0.065
	$\mu_{\Delta_z/(1+z)} \pm \sigma_{\Delta_z/(1+z)}$	-0.002 ± 0.045
	$\lvert\Delta_z/(1+z)\rvert > 0.15$	1.17 per cent

We construct data vectors corresponding to the residual Δ_z, between the true redshift z and the machine learning redshift, and also these residuals scaled by $1/(1+z)$. On these vectors we compute and compare measurements of the mean, standard deviation and the percentage of catastrophic 'outliers' defined by $|\Delta_z/(1+z)| > 0.15$. The measurements are performed on the following samples: the standard features, the top 1, and 2 best features, the standard & the top 1, 2 best features, and the standard & the top 1, 2 and 3 best features. We show the results of this analysis using Photoraptor in Table 2.

We find that the values of the mean, standard deviations and percentage of outliers decreases if we choose to use those features deemed to be the most important from Section 4.1. Furthermore, we find that the combination of the standard and top 1, 2 (or standard and top 1, 2, 3) features continue to improve each of the measured values. We find an improvement in the standard deviation of the redshift scaled residuals by 18 per cent and an improvement in the catastrophic outlier rate of 32 per cent. We note that the total training time of Photoraptor is approximately 4 h for the stated hyperparameters, on a single CPU core. The improvement in redshift estimation by combining radii, fluxes and magnitudes has been seen before (Gerdes et al. 2010), however, our choice of which additional features to use is motivated by the importance feature selection.

In the second approach, we allow the aNN architecture to vary. We calculate the machine learning redshift using the Cascade2 algorithm implemented in FANN. We choose to examine the standard feature sample and the top 1&2 feature sample. For this analysis, we marginalize over the machine architecture hyperparameters by randomizing the number of neurons in the hidden (multiply connected) layer, the learning rates, the desired best error the learning algorithm will try to attain, and the number of galaxies in the training set. We train FANN using the training set, and calculate the standard deviation of the residuals using the full test set.

We find that the value of the standard deviation of the residuals decreases by 17 ± 8 per cent using the top 1&2 features compared to the standard features. However, the value of the standard deviation for the best hyperparameter configuration is 0.081 which is not competitive with that obtained using Photoraptor.

Finally, as a further illustrative example of feature importance we determine which are the two worst, or lowest ranking, features from each iteration. The worst features are psfMag_i-psfMag_u, fracDeV_g, fiberMag_i-fiberMag_u, fracDeV_i, dered_z-dered_u, psfMag_r-psfMag_u, fracDeV_r, fracDeV_u, dered_i-dered_u, fracDeV_z. We repeat the above analysis and pass these features to Photoraptor to measure a machine learning redshift. The value of the standard deviation of the residuals is 0.089 which is larger than using the standard, or top ranked features. We note that the outlier rate is also very high, with a value of 4.5 per cent.

We present the following hypothesis which may describe why we could expect the least important features to be those listed above. Features or feature combinations involving the *u* band: above a redshift of 0.1 the 4000 Å break drops out of the SDSS *u* band. Red galaxies will no longer have a large measurement in this band which may indicate the lack of predictive power for redshifts. fracDev: this parameter describe morphology, e.g. how 'bulge-like' or 'disc-like' the galaxy is. We note that Singal et al. (2011) also show that the addition of some other morphological parameters such as concentration, does not improve the machine learning redshift.

4.3 The importance of the size of the training set

In this section, we use the computationally fast RDFs as the machine learning architecture. This allows many different training sample sizes to be examined, and the other hyperparameters to be explored in a timely manner. We sample the hyperparameter space and generate 1100 unique RDFs.

In Fig. 3, we show the effect of the size of the randomly selected training sample on the cross-validation mean μ, and standard deviation of the residuals Δ_z, in four redshift ranges (see legend). The central lines show the mean values, and the shaded regions show the error on the mean. The large star symbols and thick error bars show the mean and standard deviation values of the residuals calculated using the SDSS template-ml redshift on the same cross-validation set. The large triangles show the smallest standard deviation of the machine learning method. We have positioned the triangles next to the stars to aid comparison.

Fig. 3 shows that the bias μ_{Δ_z} on the cross-validation sample decreases as the randomly selected training sample increases until the number is ≈2000 after which it decreases less significantly and approaches a constant. The mean values are well within one standard deviation of the line $\mu_{\Delta_z} = 0$. The dispersion of μ_{Δ_z} at fixed training sample size is due to the random nature of the RDF, and also the marginalization over the RDF hyperparameters. At fixed training sample size, we find that the mean and standard deviation in each redshift bin is largely unaffected by our choices of machine architecture hyperparameters. We find that including more than $\sim$ 100 000 galaxies in the training set does not drastically improve the machine learning redshift when using the RDF method. Recall that the cross-validation sample is not shown to the machine architecture while training, and therefore presents an unbiased estimate of the error.

4.4 The effect of the machine learning framework

RDFs are constructed in Section 4.1 to measure feature importance, but they also provide a precise measurement of the machine learning redshift, as previously seen by e.g. Gerdes et al. (2010); Carrasco Kind & Brunner (2013). The hyperparameter values of the RDF are the number of trees and the minimum number of training objects on each leaf node. We fix the size of the input features to be all 85 features seen in Table 1. We randomly explore the remaining hyperparameter space. For each random instances of hyperparameters the training sample (of size up to 961 114) is used to train the RDF. For each instance, the residuals are computed on the full cross-validation sample (consisting of a different set of 480 557 galaxies).

We choose the RDF with the lowest standard deviations as our final set of hyperparameters, and finally measure the standard deviations on the residuals calculated using the test set (again, another different set of galaxies of size 480 557). We reiterate that the cross-validation data set is not used during training, and the test data set is only used in this final stage. This results in an unbiased estimate of the error when applied to new data.

The top panel of Fig. 4 shows the distribution of the test sample residuals between the true redshift and both the SDSS template-ml redshift, and the RDF redshift. We additionally show this distribution scaled by 1/(1+spec_z) by the dotted lines, and mark the mean and standard deviation of the residual distribution for each case in the legend. The bottom panel of Fig. 4 shows a redshift scatter plot using 10 000 randomly selected galaxies from the test sample. We show the spectroscopic redshift against the SDSS template-ml redshift and against the RDF machine learning redshift.

Fig. 4 shows that the distribution of residuals using the RDF method (red lines) with 85 input features is more peaked around the true redshift, and declines faster as the residuals increase, than the distribution for the same sample of galaxies using the SDSS DR10 template-ml photometric redshift (blue lines).

We find that mean and standard deviation of the residuals using the RDF machine learning redshift applied to the test sample to be

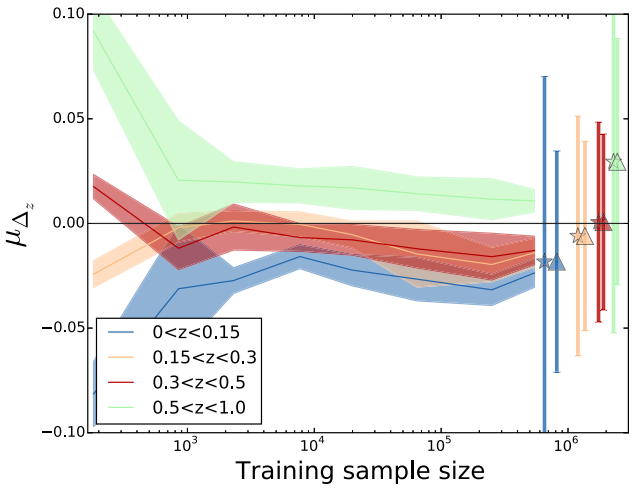

Figure 3. The effect of the size of the training sample on the mean μ value of the redshift residuals Δ_z, calculated using forests. The sample is divided into four redshift ranges (see legend) and the mean, error on the mean and the standard deviation is calculated for the residuals in each bin. The central lines show the mean values, and the shaded regions show the error on the mean. The other hyperparameters of the machine architecture are marginalized over. The large star symbols and thick error bars show the mean and standard deviation values of the residuals calculated using the SDSS template-ml redshift calculated on the same cross-validation set. The large triangles show the smallest standard deviation of the machine learning method. We have positioned the triangles next to the stars to aid comparison.

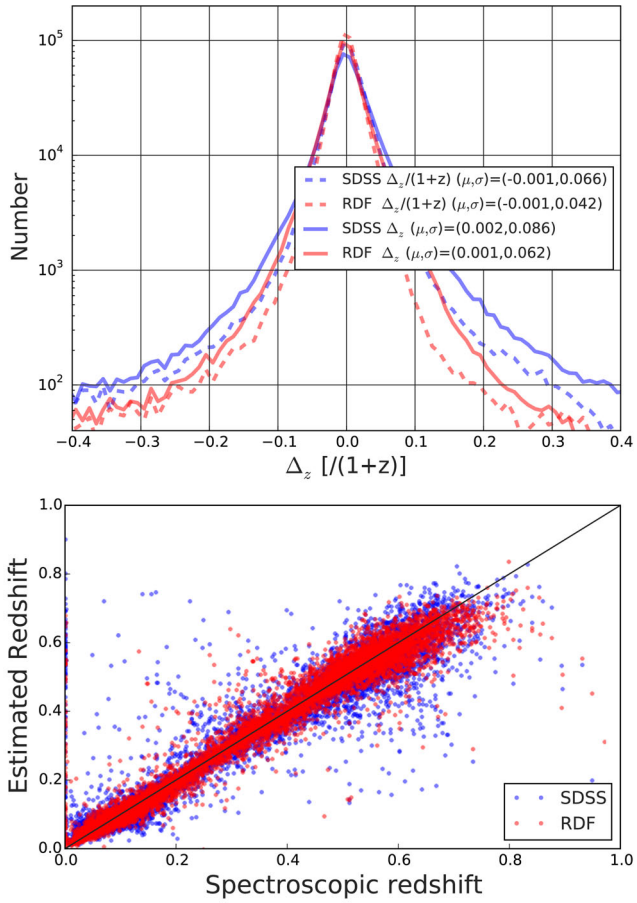

Figure 4. Top panel: the (blue) red curve shows the histogram of residuals between the true redshift and the (SDSS DR10 template-ml) RDF, redshift for the 480 557 galaxies in the test sample. The scale is logarithmic along the *y*-axis. The dotted lines show the residual distributions scaled by 1/(1+spec_z). The legend shows the values of the mean μ and standard deviation σ of each distribution. The test sample is not used during the RDF training or hyperparameter selection. The hyperparameter values of the RDF can be found in the text. Bottom panel: a scatter plot showing 10 000 randomly selected galaxies from the test sample. We show the spectroscopic redshift against the SDSS template-ml redshift and against the RDF machine learning redshift.

−0.001 and 0.061, and for the residuals scaled by 1/(1+spec_z) we measure −0.003 and 0.041. We next calculate the mean and standard deviation of the residuals on the test set using the SDSS template-ml photometric redshift and find them to be 0.002 and 0.086, and for the residuals scaled by 1/(1+spec_z), we measure −0.001 and 0.066. These values are also shown in Fig. 4. We note that the mean and standard deviations of the residuals of the RDF machine learning redshift are comparable, but slightly better than those measured using the aNN in Table 2. We reiterate that the RDF has used the full set of 85 input features, whereas the aNN were restricted to the standard, or top 1, 2, 3 most important features. A more complete analysis of the ability of an aNN to measure a machine learning redshift with a much larger input features is planned for future work.

We next restrict the input features of the RDF to be the 'standard' features defined above, we measure the mean and standard deviation of the residuals Δ_z to be −0.002 and 0.075. This results in a slight improvement over the SDSS template-ml method, but is not as powerful as using the full 85 input features. We find that the outlier rate using the RDF method and the standard features is 1.74 per cent.

To summarize, we find that the best-fitting RDF machine learning redshift using the full 85 input feature list has a decrease in the standard deviation of 29 per cent (38 per cent) for the residual distribution (for the residual distribution scaled by 1/(1+spec_z)) when compared to the SDSS DR10 template-ml photometric redshifts. Following Hildebrandt et al. (2010), we define the outlier rate as $|\Delta z/(1+\text{spec\_z})| > 0.15$, and find that the outlier rate of the galaxies in the test sample using the SDSS DR10 template-ml redshift to be 2.32 per cent and using the RDF redshift method the value is reduced to 0.99 per cent, which is a 57 per cent reduction. We note that the hyperparameters values of the final RDF are *Number of training examples: 707 553, Minimum number of examples per leaf: 6, Number of trees in the forest: 55*. We also note that the most important ranked 1, 2, 3 features of final RDF are fiberMag_g, fiberMag_i and the colour dered_z-dered_g. We make the data sets and best trained RDF available on the homepage of the lead author upon journal acceptance.

5 CONCLUSIONS

To exploit large-scale photometric surveys the identification of galaxies and measurement of their positions on the sky and in redshift space is paramount. Very accurate spectroscopic redshifts z, can only be measured on a small subset of galaxies due to the integration times required to obtain a reliable measurement.

One may determine a less accurate distance measurement using the photometric properties of the galaxy using either template methods, which encode our parametric knowledge of stellar populations and redshift expansion, or machine learning methods which are non-parametric. In machine learning methods, photometrically measured or derived properties, or 'features' are chosen and presented to the machine learning architecture in the hope that the input features identify a good scaling with redshift.

In this paper, we present a study of the effect on the recovered machine learning redshift of the choice of input photometric features. We select and derive 85 easily obtained photometric features of all galaxies with spectroscopic redshifts found in SDSS DR10 CasJobs. We apply very light quality cuts on the recovered galaxies and obtain a sample of 1.9 million galaxies.

We use the machine learning architecture Decision Trees combined into Forests (RDF) with ADABOOST (Breiman et al. 1984; Drucker 1997; Freund & Schapire 1997) learning to perform feature importance using the Gini criteria to determine which features produce the most predictive power for redshift estimation, and find that the SDSS *g*-band fibre magnitude is the top ranked, best single feature in 67 per cent of cases. We list the top 1, 2, 3 features, and show their occurrence fractions after randomly exploring RDF hyperparameters in Fig. 1. ADABOOST and RDFs have been used previously to estimate machine learning redshifts but only using standard magnitudes as input features (e.g. Gerdes et al. 2010).

We show how the addition of the top 1, 2, 3 importance ranked features can improve machine learning redshift estimates using the aNN Photoraptor and FANN. Photoraptor allows a two layer deep network to be trained efficiently, and FANN implements Cascade2 learning which sequentially adds (multiply connected) hidden neurons to the hidden layer.

We continue by fixing the Photoraptor machine architecture and present it with the standard and then higher dimensional input features. We find that the recovered machine learning redshift is improved with the addition of important features. This is in agreement

with Tagliaferri et al. (2003) who show that the addition of extra features such as radii and fluxes improve the machine learning redshift estimation compared with using just magnitudes alone. However, note that Singal et al. (2011) use more elaborate morphology features and find no real improvements to the machine learning redshifts. In this work, we decide which additional features to present to the aNN using the feature importance selection.

We present the standard and top ranked 1, 2 important features to FANN and allow the machine architecture to vary. While the absolute results are not as competitive as with Photoraptor, the use of the top 1, 2 important features does improve the redshift estimate.

To quantify the improvement of the machine learning redshift, we measure the residual between the true spectroscopic redshift and the machine learning redshift Δ_z. We measure the mean and the standard deviation of Δ_z and the fraction of catastrophic 'outliers' defined by $|\Delta_z/(1+z)| > 0.15$. We find that the addition of the top ranked features to the standard features decreases the standard deviation of the residuals by 13 per cent, and by 18 per cent for the residual normalized by $1/(1+z)$, and that the outlier fraction is also decreased by 33 per cent. We reiterate that this improvement is due to the addition of easily obtainable photometric features. When using FANN, we note that the standard deviation of the residuals decreases by 17 ± 8 per cent after marginalizing over the aNN hyperparameters.

We then show how RDFs can also determine a machine learning redshift (see also Gerdes et al. 2010; Carrasco Kind & Brunner 2013), and document that this technique uses less computational resources, decreases the standard deviation of the residuals *and* lowers the outlier fraction compared with the aNN architectures implemented here.

We quantify the improvement using RDFs and all 85 input features, with respect to the SDSS DR10 photometric redshift available from CasJobs, and with respect to the two aNN architectures. We find that the standard deviation of the residuals distribution decreases by ≈29 per cent, and by ≈38 per cent for the distribution scaled by $1/(1+z)$. We show that the outlier rate of the galaxies in the test sample using the SDSS DR10 photometric redshift is 2.32 per cent and using the RDF method the value decreases to 0.99 per cent.

We note that other machine learning architectures are readily available and the use of additional features within these frameworks is an ongoing project. Given the non-parametric nature of the machine learning described here, we caution that the results of this analysis are not necessarily easily transported to other surveys or data sets. It would be prudent to perform a similar analysis with different surveys in order to identify their most salient features. However, this problem is very tractable. Using Decision Trees and boosting routines, one is able to analyse a data set of millions of galaxies in just a few hundred seconds using a single core machine.

ACKNOWLEDGEMENTS

We would like to thank the referee for useful comments and suggestions which have improved the paper. Funding for the SDSS and SDSS-II has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, the US Department of Energy, the National Aeronautics and Space Administration, the Japanese Monbukagakusho, the Max Planck Society, and the Higher Education Funding Council for England. The SDSS website is http://www.sdss.org/.

REFERENCES

Abazajian K. N. et al., 2009, ApJS, 182, 543
Ahn C. P. et al., 2012, ApJS, 203, 21
Ahn C. P. et al., 2014, ApJS, 211, 17
Breiman L., Friedman J. H., Olshen R. A., Stone C. J., 1984, Classification and Regression Trees. Wadsworth International Group, Belmont, CA
Brescia M., Cavuoti S., D'Abrusco R., Longo G., Mercurio A., 2013, ApJ, 772, 140
Brescia M. et al., 2014a, PASP, 126, 783
Brescia M., Cavuoti S., Longo G., De Stefano V., 2014b, A&A, 568, A126
Budavári T., Szalay A. S., Connolly A. J., Csabai I., Dickinson M., 2000, AJ, 120, 1588
Carrasco Kind M., Brunner R. J., 2013, MNRAS, 432, 1483
Cavuoti S., Brescia M., Longo G., 2014, preprint (arXiv:1406.3192)
Collister A. A., Lahav O., 2004, PASP, 116, 345
Connolly A. J., Csabai I., Szalay A. S., Koo D. C., Kron R. G., Munn J. A., 1995, AJ, 110, 2655
Csabai I., Dobos L., Trencséni M., Herczegh G., Józsa P., Purger N., Budavári T., Szalay A. S., 2007, Astron. Nachr., 328, 852
Dahlen T. et al., 2013, ApJ, 775, 93
Drucker H., 1997, in Proc. 14th International Conference on Machine Learning ICML '97, Improving Regressors using Boosting Techniques. Morgan Kaufmann Publishers, San Francisco, CA, p. 107
Eisenstein D. J. et al., 2011, AJ, 142, 72
Fahlman S. E., Lebiere C., 1990, Advances in Neural Information Processing. Morgan Kaufmann Publishers, San Francisco, CA, p. 524
Feldmann R. et al., 2006, MNRAS, 372, 565
Freund Y., Schapire R. E., 1997, J. Comput. Syst. Sci., 55, 119
Gerdes D. W., Sypniewski A. J., McKay T. A., Hao J., Weis M. R., Wechsler R. H., Busha M. T., 2010, ApJ, 715, 823
Gunn J. E. et al., 2006, AJ, 131, 2332
Hildebrandt H., Arnouts S., Capak P., Moustakas L. A., Wolf C., Abdalla e. a., 2010, A&A, 523, A31
Ilbert O. et al., 2009, ApJ, 690, 1236
Lahav O., 1997, in Di Gesu V., Duff M. J. B., Heck A., Maccarone M. C., Scarsi L., Zimmerman H. U., eds, Data Analysis in Astronomy. World Scientific, Singapore, p. 43
Li L., Zhang Y., Zhao Y., Yang D., 2006, preprint (arXiv:astro-ph/0612749)
Nissen S., 2003, Technical report, Implementation of a Fast Artificial Neural Network Library (fann). Department of Computer Science, University of Copenhagen (DIKU)
Nolan L., Ani R. T., 2008, Comput. Sci. Eng., 10, 18
Pedregosa F. et al., 2011, J. Mach. Learn. Res., 12, 2825
Sánchez C. et al., 2014, MNRAS, 445, 1482
Singal J., Shmakova M., Gerke B., Griffith R. L., Lotz J., 2011, PASP, 123, 615
Smith J. A. et al., 2002, AJ, 123, 2121
Tagliaferri R., Longo G., Andreon S., Capozziello S., Donalek C., Giordano G., 2003, in Apolloni B., Marinaro M., Tagliaferri R., eds, Lecture Notes in Computer Science, Vol. 2859, Neural Nets. Springer-Verlag, Berlin, p. 226
Yeche C. et al., 2009, preprint (arXiv:0910.3770)
Yip C.-W., Szalay A. S., Carliles S., Budavári T., 2011, ApJ, 730, 54
York D. G., SDSS Collaboration, 2000, AJ, 120, 1579

This paper has been typeset from a TEX/LATEX file prepared by the author.