MAPPING THE UNIVERSE: USE MACHINE LEARNING TO FIND THE DISTANCES TO FARAWAY GALAXIES

MSC Project Report

Student: Nnamdi Chimezie
Supervisor: Dr. Marika Asgari

ABSTRACT

Redshift denoted as *Z* is an important phenomenon in astronomy as it provides a measurable distance to faraway galaxies. We can derive redshift by fitting the observed photometry of a galaxy to a set of templates through spectroscopy. Redshift can be measured accurately using spectroscopy but this method is expensive, instead we use photometry. The data used for this research is based on the cosmological simulation called Multiple Imputation by Chained Equation MICE (Fosalba et al., 2015) and contains about 202046 galaxies. A Kilo-Degree Survey (KiDS) magnitude limit of $r < 19.87$ was applied to the dataset to select the brightest galaxies which would make this sample almost identical to data from Galaxy and Mass Assembly survey (GAMA). Several machine learning (ML) algorithms were used to train various models from samples of galaxies that has spectroscopy and used to predict the redshift for galaxies that do not have spectra. The average of the redshift distribution was 0.81% with the Random Forest model having the highest accuracy of 0.93 using the co-efficient of determination (R^2) as the metric. Hyperparameter tuning was done on some models deployed and was shown to improve the accuracy of the KNN model by 3%.

INTRODUCTION AND BACKGROUND

There are several probes used in cosmology such as gravitational lensing, weak lensing, and galaxy clustering for which we need to know the distances to faraway galaxies and redshift provides a measurable distance (Weinberg et al., 2013). Ferguson et al., 2004 described redshift as measure of how an object in space is moving relative to us. When a star is moving away from us, due to the doppler effect, the light from the star is of a longer wavelength and thus moves towards the red end of the spectrum. Hence, the more distant a galaxy is from us, the more its light will be redshifted.

Edwin Hubble discovered this phenomenon in 1929 and since then has been applied to deep-field and wide-field surveys like the Hubble Deep Field (Fernandez et al., 1999) and the Sloan Digital Sky Survey (SDSS; Csabai et al., 2003).

Redshifts can be estimated using photometry which is the measuring of light as relates to its perceived brightness. Photometry are being done in passbands through the use of optical filters (Bessell, 2005). Photometric redshifts (*photo-zs*) are redshift estimates of galaxies based on observations like magnitudes or colour and can be useful for some studies (Mandelbaum et al.

2008). We can derive redshifts by fitting the observed photometry of a galaxy to a set of templates through spectroscopy (Cimatti et al., 2002).

Previous literature on this case study used *ANNz2*, which is a public software package available on the GitHub repository (Github, 2022) to estimate *photo-zs* (Sadeh et al., 2016). Collister & Lahav, 2004 found out that *ANNz* is useful as it understands the relation between photometry and redshift from an appropriate training set of galaxies for which the redshift is already known. Also, Bilicki et al., 2021 used supervised machine learning neural networks algorithm implemented in the ANNz2 software to select *photo-zs* from bright galaxy samples in the Kilo-Degree Survey (KiDS) data (Kuijken et al., 2015). They found out that the redshift combined with the 9-band photometry can estimate absolute magnitudes and stellar masses for the full sample. Kügler et al., 2015 in his work reviewed the use of KNN in determining spectroscopic redshifts while Wright et al., 2020 talked about photometric redshift calibration using self-organizing maps (SOM) which is another type of neural network. Neither of the papers above used other machine learning algorithm to estimate *photo-zs.*

Getting the estimate of redshifts for every galaxy using spectroscopy is difficult as it involves a lot of time and is also expensive which is why we use photometry. Using simulated data, we will compare multiple machine learning models used in other fields to calibrate redshift and see if we can get similar or better results to the ones used above.

DATASET

The data used for this research is the KiDS (UGRI) and VISTA Kilo-degree Infrared Galaxy Survey (VIKING-ZYJHK$_S$, Edge et al., 2013) redshift calibration validated on the MICE cosmological simulation (Fosalba et al., 2015) which contains about 202046 galaxies. Simulated data was used so we can evaluate the accuracy of the model since we already know what the true redshifts are. Astropy (Robitaille et al., 2013) was used to read in the dataset and some visualizations were made. Exploratory data analysis was performed on the dataset and it was found out that there were no missing values or empty columns and as such data cleaning wasn't a required practice.

The data contains different observational characteristics such as galaxy right ascension and declination, observed galaxy redshifts which is the true redshift, KiDS 9-band Bayesian Photometric Redshift (BPZ) which represents the estimated redshift, their magnitudes (UGRIZYJHK$_s$) which represent the colours- this enables us to know the redshift of galaxies by showing how bright the galaxies are in each filter- their photometric noise and also the lens weight.

For optimal completeness, KiDS bright samples were selected by applying a magnitude limit of $r < 19.87$ to the dataset resulting in pure bright samples of galaxies that matches the highly complete spectroscopic data from Galaxy and Mass Assembly survey (GAMA, (Driver et al., 2011).

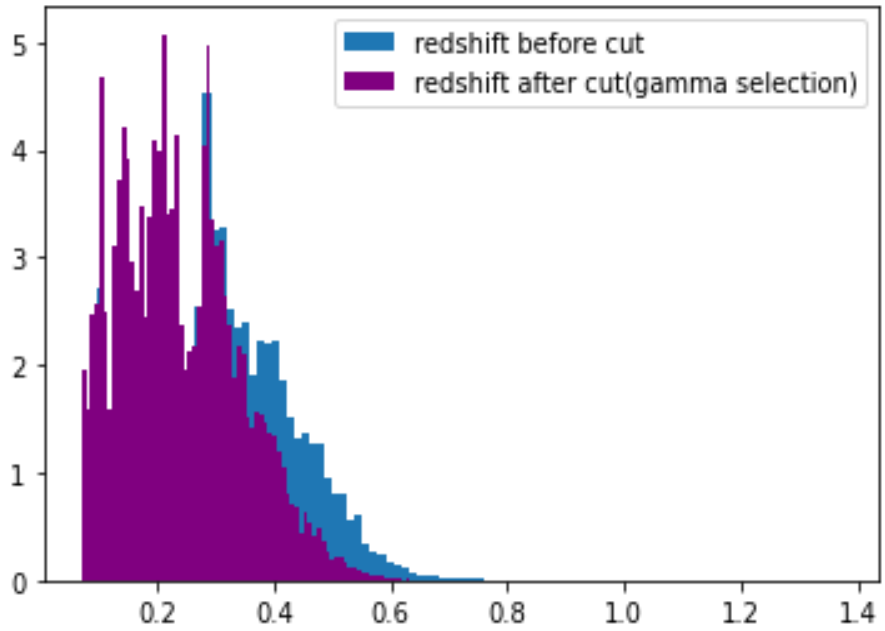


FIG 1- Effect of the r-band cut on the redshift

Figure 1 shows the output after a magnitude limit of r <19.8 was applied to select bright galaxies thereby making it look like GAMA which is the spectroscopic observations that we are interested in. This resulted in bright galaxies that matches GAMA being retained and faint galaxies between 0.6 and 0.8 being filtered out.

METHODOLOGY

Since we are trying to understand the functional relationships between continuous dependent variable and an independent variable, we use regression analysis for data processing (Isobe et al., 1990). Five regression models were applied and their performances were compared against each other. Before deploying the models, data pre-processing was done using the following steps:

- Separated the dependent variable from the independent variable.
- Normalized the data between 0 and 1.
- Split the data into training and testing. As a general rule of thumb, I used 80% for training and 20% for testing.

After this was done, the various models used were then fitted to the training data and used to validate the test dataset. K-fold cross-validation- which is a resampling procedure- was used to evaluate the various models deployed. K refers to the number of groups that the dataset is split into. The metrics used to evaluate the models are Mean squared error (MSE), Mean absolute error (MAE) and the Co-efficient of determination (R^2). To avoid error generalisation, I didn't evaluate the model performance on the test sample (Goodfellow et al., 2016).

MAE = $(1/n) * \Sigma|y_i - x_i|$ Σ= summation notation, y_i= predicted values, x_i= true values, n= number of observations

MSE formula = $(1/n) * \Sigma (y_i - x_i)^2$ n = number of observations, Σ = summation notation, y_i = true values, x_i = predicted values

$R^2 = 1- (RSS/TSS)$ RSS = sum of squared residuals, TSS = total sum of squares

MODELS USED

Random Forests

Random Forest is an ensemble statistical technique that uses bagging or bootstrap aggregation to merges the decisions of different multiple trees in order to find an answer which represents the average of all the decision trees in the case of regression. Breiman, 1996 developed the random forest algorithm to address the problem of overfitting and reduce the variance which is a major disadvantage in decision trees.

Since its inception it has been applied to various fields including astronomy (Chen et al., 2022) and has been used to detect fraud in banks accounts (Kaggle, 2022). I chose this model for this regression task due to its high accuracy and flexibility.

70% of the training set was fit to the model on default parameters and used to make predictions on the test set. The model was then evaluated using the metrics mentioned above.

K-Nearest Neighbour KNN

KNN is a non-parametric supervised machine learning algorithm that can be used to make predictions on a given dataset. It was first developed by Fix & Hodges, 1989 and later expanded by Altman, 1992 . KNN has been applied in previous study. Hildebrandt et al., 2021 used KNN for redshift estimation. Also Kügler et al., 2015 determined spectroscopic redshift by using KNN regression. It is based on the idea that the observations nearest to a datapoint *K* are the most similar observations in the dataset (TDS, 2022).

75% of the data was used to train the KNN regressor with default parameters and *K* value set to 2. *K* represents the number of nearest points that will be checked. The values from the metrics used to evaluate the model was then recorded.

Support Vector Machines (SVM)

SVM is a supervised machine learning ML algorithm that can be used to solve the regression problems such as the one we are interested in. It was developed by Cortes & Vapnik, 1995 at AT&T Bell Laboratories.

The SVM algorithm takes datapoints which are referred to as support vectors to construct a hyperplane (Ding et al., 2014). The dimension of the hyperplane is determined by the number of features. This model was chosen because of its easy implementation and ability to detect outliers.

70% of the dataset was used to train the model and 30% was reserved for the test set. The model was trained on default parameters with the radial basis function (RBF) kernel and evaluated on the validation set.

Neural Networks

Neural networks are one of the most popular algorithms in Machine Learning that covers a broad range of concepts and techniques. It Is commonly referred to as a black box algorithm because it can be hard to understand what they are doing. According to IBM, 2022 neural networks are a computerized modification of the interconnected biological neurons in the brain consisting of several node or artificial neurons organized into layers allowing computer programs to recognize patterns and solve common problems in the fields of AI, machine learning, and deep learning. Collister & Lahav, 2004 and Wright et al., 2020 both used Artificial neural network (ANN) and self-organizing maps (SOM) to calibrate photometric redshifts.

I trained a feedforward multilayer perceptron (MLP) which is a type of neural network consisting of an input layer and few hidden layers and an output layer. Data is relayed from the input layer which receives information contained in the dataset to the output layer which gives the final result after the data has passed through all the layers. In-between the input layer and the output layer are the hidden layers where the artificial neurons are contained. The network learns by forming probability-weighted associations within the data structure of the neural net. The various weights between the nodes determines how much influence each input has on the output. The higher the weight the more influence that unit has in another.

Due to its popularity and decision-making ability neural networks are widely applied in various fields like cyber security to detect fraudsters, natural language processing (NLP), Google's search algorithm, facial recognition software, stock market prediction, etc. I chose to work with this model due to its wide application and efficiency in predicting data.

70% of the dataset was used during training. Adaptive moment estimation (ADAM) optimizer and Rectified linear unit (ReLU) activation function were the selected hyperparameters. After 10 epochs of training the data with a batch size of 32, the results were then recorded using the evaluation metrics above. Model tuning was then performed to see if higher accuracy can be achieved.

RESULTS

Correlation Matrix

A correlation matrix which displays the correlation coefficients for each variable was visualised to measures the degree of linear relationship between each pair of variables in the dataset.

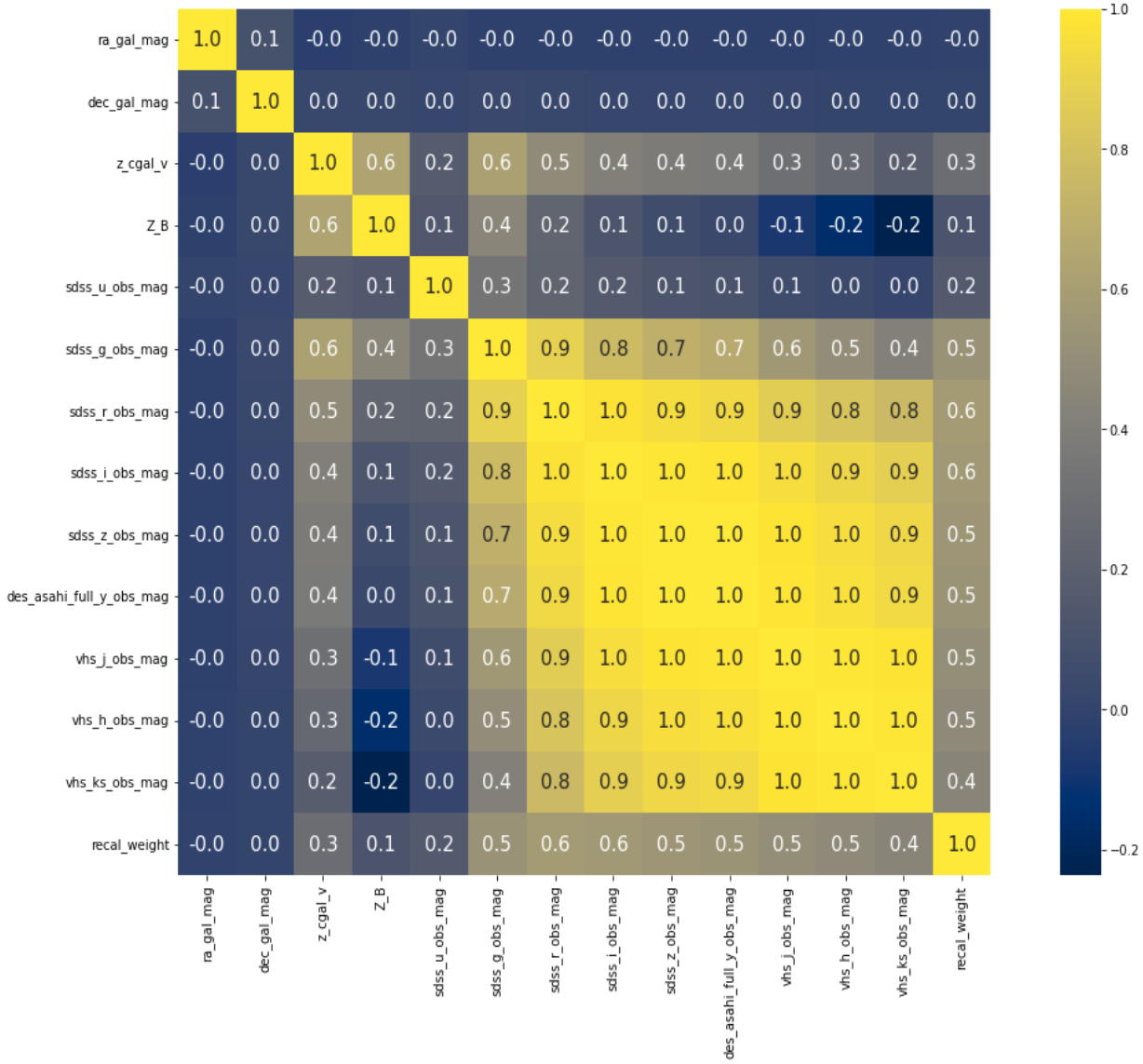


Fig 2: Correlation matrix

From the matrix above the most important correlations are the ones with the true redshift which is what we are modelling. The true redshift has various levels of correlation with the data- the highest value being 0.6 with the estimated redshift. This tells us that the data might be learning something from the estimated redshift, thus it would be interesting to see how the data responds if this column is dropped in further study. The galaxy right ascension and declination on the other hand-which are like latitude and longitude representing their position in the sky-has no correlation with the data and can be dropped off without affecting the results.

Redshift Distribution

FIG 3- Bayesian Probability estimation

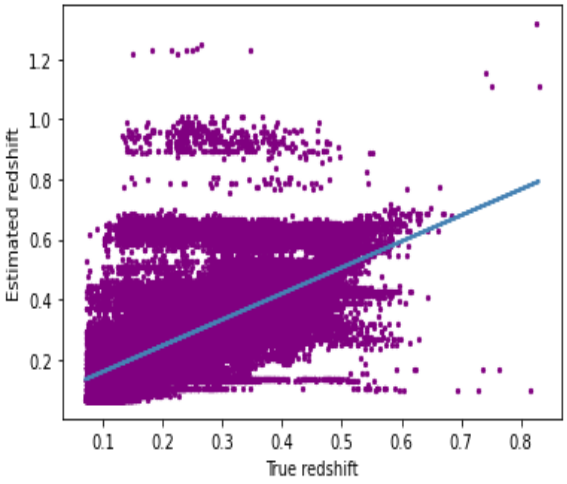


FIG 4- Random Forest

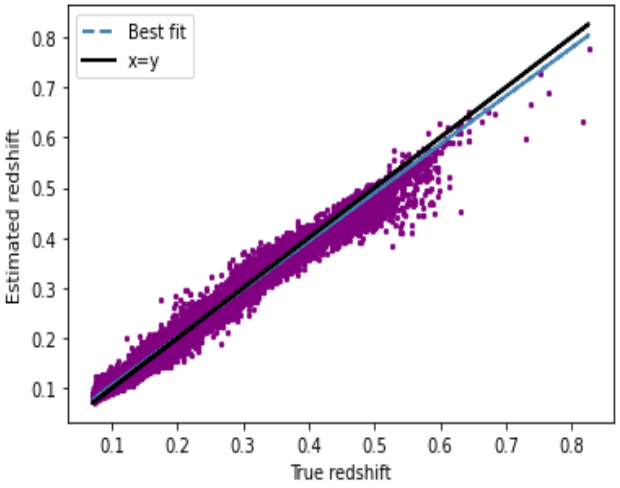


FIG 5- Linear regression

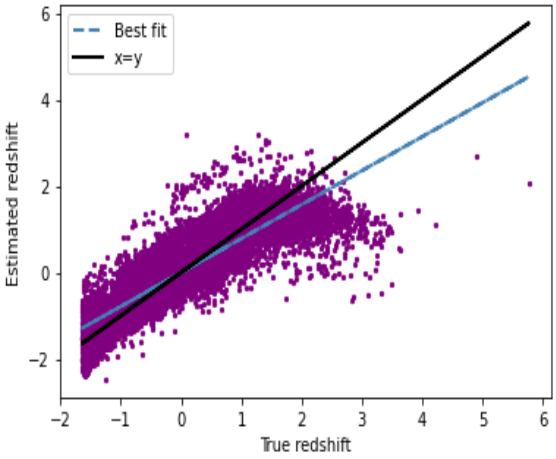


FIG 6- SVM

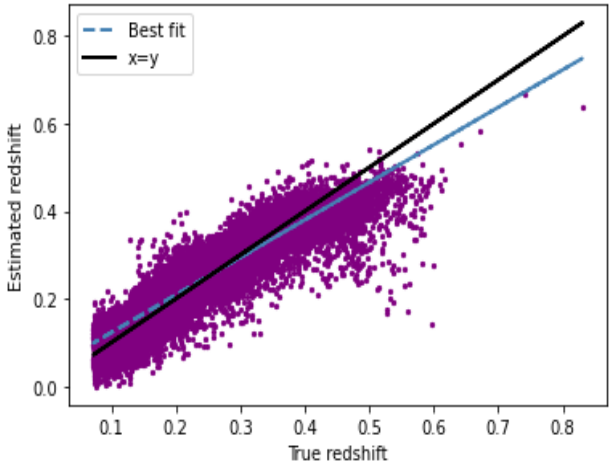


Fig 7- KNN

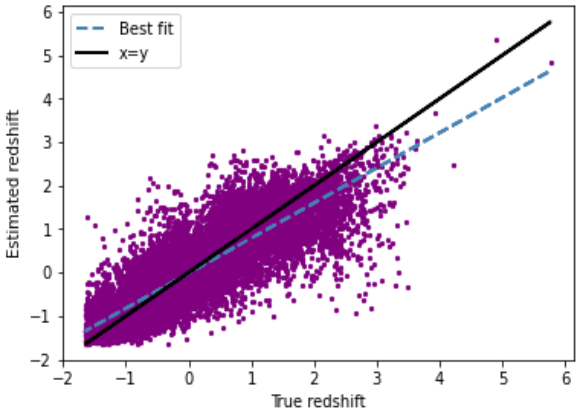
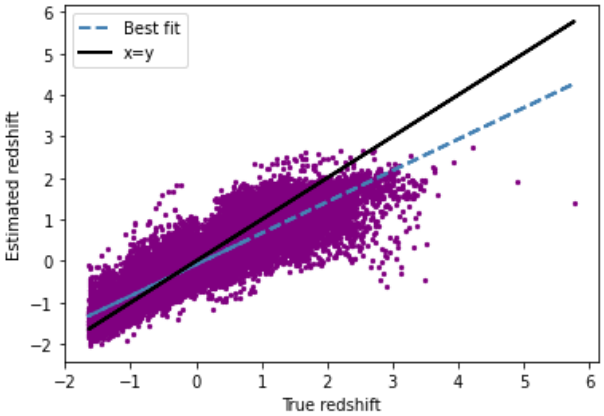


Fig 8- Neural Network



Figures 3-8 shows the relationship between the estimated and true redshift for the various models deployed. The aim is to have a line of best fit that is as close as possible to the x=y line and contains points with minimal scatter around this line as possible.

Fig 3 shows the distribution between the true and estimated redshift using BPZ estimation. The BPZ model's line of best fit is further away from the line x=y and have many points scattered around the line.

As can be seen in Fig 4 the random forest best fit model is as close as possible to the truth line x=y which is also reflected in its performance level (see table 1). Although redshift around datapoint 0.5 and 0.6 appears to coming down, this could be as a result of some degeneracies in colour space as we have a limited number of colours. Perhaps, if we had more photometric bands, then we might be able to map where these degeneracies belong to and better fit our model.

From figure 5, the linear regression line of best fit model is further away from the x=y line with so many scattered points along the line and this can be seen in the model's performance (see table 1).

Figure 6 shows the SVM line of best fit model distant from the x=y line with the datapoints scattered along this line. This is reflective of the model's performance in table 1.

As seen in figure 7 the KNN line of best fit model has many points scattered along this line and is not as close as possible to the truth line x=y.

As can be seen in Fig 8 the neural network best fit model is the furthest away from the truth (x=y) which is also reflected in its performance level (see table 1). It also has so many points scattered along the line.

MODELS PERFORMANCE

Model	**Accuracy (%)**	**K-Fold Mean**	**Standard Deviation**	**Mean Difference (y_true – predicted)**	**Time**
Linear regression	77%	0.77	0.005	0.0003	2.3 sec
KNN	76%	0.76	0.006	0.01	4 sec
Support Vector Machines	83%	0.83	0.004	0.002	24 sec
Random Forest	93%	0.93	0.003	0.0001	1m 24sec
Neural Network	79%	N/A	N/A	0.085	1m 28sec

Table 1: Shows accuracy achieved across the 5 models deployed

Table 1 shows the performance of the various models deployed accuracy after evaluating on the validation data. The accuracy was estimated using the Co-efficient of determination R^2 (see table 2). The models did not perform badly as they were able make predictions based on the information contained in the dataset.

The random forest regressor achieved the highest accuracy and this could be attributed to the fact that the model didn't overfit to the training data due to the large number of relatively uncorrelated trees in the random forest model. The trees protect each other from their individual errors leading to low variance and better accuracy.

The K-fold mean achieved across the accuracy of the 4 models is within range of the final result in each regressor. Hence, we can say the results achieved are not stochastic; thus, we can rely on them.

The standard deviation and mean difference are all within 0.01 with random forest model achieving the lowest. This tells us that the mean of the predicted output is as close as possible to the mean of the real value, which means the dataset did not contain galaxies with inordinately large figures as this can affect the measure of dispersion in the dataset. Also, it can be used in outlier detection.

Random forest took the longest time to train than the rest model. This is because of the number of trees involved and is actually a trade off with accuracy.

EVALUATION METRICS

Metrics	Linear Regression	KNN	SVM	Random Forest	Neural Network
MAE	0.33	0.34	0.03	0.01	0.33
MSE	0.22	0.23	0.001	0.0006	0.20
R^2 score	0.77	0.76	0.83	0.93	0.79

Table 2: Shows the various metrics used to evaluate the model accuracy

Mean absolute error (MAE) is the absolute value of the differences between the predicted and actual values. It gives you an idea of the errors contained in the model predictions. Random forest model has the lowest error of 0.01. This means that each prediction in the random forest model is 0.01 lesser or greater than the actual prediction.

Mean squared error gives us the average of all the square of the differences between the predicted and actual values. It is generally lower than the MAE and it amplifies differences i.e., a large difference in predicted value can cause the value of the MSE to be greater than the MAE. From the table 2 we can see that the MSE of all the model used are below the MAE which tells us that there are no outliers or inordinately large figures.

R^2 score or the coefficient of determination represents the proportion of the variation in the dependent variable that is predictable from the independent variable. It tells us how well the regression model fits to the observed data. The closer the value is to 1.0 the better and more reliable is the model. The downside of R^2 is that it does not exactly tell us how badly the model is in terms of the errors in the prediction. From the above, random forest gave us the closest

value to 1.0 which is 0.93, followed by SVM with an R^2 of 0.83. KNN and linear regression has similar score with 0.76 and 0.77 respectively.

HYPERPARAMETER TUNING

Hyperparameter tuning was done on 3 selected models as it was difficult to carry out optimization on all the models deployed due to the large training time involved. The aim of optimisation is to find the best possible combination of parameters to see if we can achieve a higher accuracy than on default parameters.

Model	Method	Accuracy	Time	Parameter Space
Random forest	Random Search CV	91%	303m	max depth= 100, number of estimators= 400
	Grid Search CV	86%	8m	max depth= 10, number of estimators= 15
KNN	Grid Search	79%	1m 11sec	number of neighbours=8
SVM	Grid Search	70%	13m	
Neural Network	Manual	104%	106m	Batch size = 15, epochs = 5, optimizer= Adam
	Grid Search	70%	192m	Optimizer= Rmsprop, batch size= 30, epochs= 10

Table 3: Hyperparameter Optimization

As seen in table 3, when using random search CV best parameters on the random forest model we got an accuracy of 91% which is lesser than the accuracy got on default parameters (see table 1). On using grid search we got an even lower accuracy of 86% after training the model on the optimized parameters.

For the KNN model, the grid search method gave us an accuracy of 79% which was an increase from to the previous accuracy which we recorded as 76%.

The SVM saw a drop in accuracy to 70% after grid search method was applied.

The neural network was the most difficult to tune as it took a long time to train the model. An accuracy of over 100% was gotten using the manual approach. This is an arbitrary value as accuracy cannot be more than 100%. Using grid search CV, we achieved an accuracy of 70% which is a drop from the initial accuracy recorded before the model was tuned.

Tuning didn't make much of a difference on the model's accuracy as in most cases the accuracy was low or remained the same as in the case of random forest. This could be to a number of factors like overfitting, too few parameters, using the wrong metric, trusting default values, grid search, random search, etc. Unfortunately, all these cannot be covered within the scope of this project and will leave room for further study.

DISCUSSION

This research portrays the use of machine learning to find the distances to faraway galaxies. Several algorithms were trained to the MICE data (Fosalba et al., 2015) which closely resembles the complete spectroscopic data from the fourth public KiDS data release (Kuijken et al., 2015). We applied a magnitude limit of $r<19.8$ to the data. This resulted in pure bright galaxy samples that closely resembles the properties of GAMA.

The redshift distribution in the dataset has a mean of $z= 0.23$. This tally with the findings from of (Bilicki et al., 2021) where he found the mean difference of the redshift to be 0.23.

The random forest ML algorithm performed better than the rest of the model and gave us better values based on the various metrics used to evaluate the model. This correlates with the findings from (Sadeh et al., 2016) who used Boosted Decision Trees (BDT) which is like a variation to the random forest model but contains Probability Distribution Function (PDF). He found out that the PDF gave a better estimation of the true redshift than the ANNz which is a public photometric estimation software.

Using the random forest regressor, we evaluated the R^2 score to be 93% with mean difference of $z= 0.001$ between real and predicted values. The mean scatter of the random forest model was found to be 0.003 as compared to 0.018 found in Bilicki et al., 2021.

From figure 4, the line of best fit was as close as possible to the true $x=y$ line with very minimal points scattered along the line. The random forest model was great at minimizing the differences between the predicted and actual values for the redshift.

Overall, the results from the random forest model can be relied on as it was similar in comparison to other related works. I recommend using the random forest model for this kind of problem in future studies as it has shown it is better at estimating redshift due to the large number of trees involved in the split. This helps in preventing overfitting and also minimizes the error in each individual split. The only downside to this model is that it is time consuming and takes a while to fit to the training data.

CONCLUSION

In conclusion, this research has shown that the effectiveness of the random forest model in estimating redshifts. The project also explored the use of other machine learning algorithms and found out that the KNN was not suitable for this task as it gave us the lowest accuracy using the evaluation metrics (see table 2).

Further study on this line of research should focus on using ANNz2- a public software package available on the GitHub repository (Github, 2022) for estimating photometric redshifts

(Collister & Lahav, 2004)- to evaluate this data and see if it can achieve a better accuracy than the one used here. The neural network used didn't perform as expected and there wasn't enough time in tuning the hyperparameters due to the long training time involved in using grid search.

In the future, the performance of the random forest model should be evaluated on real-world data to see how this model does at finding redshift from the GAMA. Lastly, there is a trade-off between accuracy and hyperparameter tuning. Further study may find it useful to investigate this and see if an accuracy of 1 can be achieved without overfitting the data.

REFERENCES

Altman, N. S. (1992) An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician,* 46 (3), 175-185.

Bessell, M. S. (2005) Standard photometric systems. *Annual Review of Astronomy and Astrophysics,* 43 (1), 293-336.

Bilicki, M., A. Dvornik, H. Hoekstra, A. H. Wright, N. E. Chisari, M. Vakili, M. Asgari, B. Giblin, C. Heymans, H. Hildebrandt, B. W. Holwerda, A. Hopkins, H. Johnston, A. Kannawadi, K. Kuijken, S. J. Nakoneczny, H. Y. Shan, A. Sonnenfeld & E. Valentijn (2021) Bright galaxy sample in the kilo-degree survey data release 4. *Astronomy and Astrophysics (Berlin),* 653.

Breiman, L. (1996) Bagging predictors. *Machine Learning,* 24 (2), 123-140.

Chen, S., Sun, W. & He, Y. (2022) Application of random forest regressions on stellar parameters of A-type stars and feature extraction. *Research in Astronomy and Astrophysics,* 22 (2), 025017.

Cimatti, A., Mignoli, M., Daddi, E., Pozzetti, L., Fontana, A., Saracco, P., Poli, F., Renzini, A., Zamorani, G. & Broadhurst, T. (2002) The K20 survey-III. photometric and spectroscopic properties of the sample. *Astronomy & Astrophysics,* 392 (2), 395-406.

Collister, A. & Lahav, O. (2004) ANNZ: Estimating photometric redshifts using artificial neural networks. *Publications of the Astronomical Society of the Pacific,* 116 (818), 345-351.

Cortes, C. & Vapnik, V. (1995) Support-vector networks. *Machine Learning,* 20 (3), 273-297.

Csabai, I., et al. (2003) *Astronomical Journal* 125, (p. 580).

Ding, S., Hua, X. & Yu, J. (2014) An overview on nonparallel hyperplane support vector machine algorithms. *Neural Computing and Applications,* 25 (5), 975-982.

Driver, S. P., Baldry, I. K., Bamford, S. P., Hopkins, A. M. & Liske, J. (2011) Galaxy and mass assembly (GAMA): The GAMA galaxy group catalogue (G3Cv1). *Monthly Notices of the Royal Astronomical Society,* 416 (4), 2640-2668.

Edge, A., Sutherland, W., Kuijken, K., Driver, S., McMahon, R., Eales, S. & Emerson, J. P. (2013) The VISTA kilo-degree infrared galaxy (VIKING) survey: Bridging the gap between low and high redshift. *The Messenger,* 154 32-34.

Ferguson, H. C., Dickinson, M., Giavalisco, M., Kretchmer, C., Ravindranath, S., Idzi, R., Taylor, E., Conselice, C. J., Fall, S. M. & Gardner, J. P. (2004) The size evolution of high-redshift galaxies. *The Astrophysical Journal,* 600 (2), L107.

Fernandez-Soto, A., Lanzetta, K. M., and Yahil, A. (1999) *Astrophysical Journal* 513, (p. 34).

Fix, E. & Hodges, J. L. (1989) Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale De Statistique,* 57 (3), 238-247.

Fosalba, P., Crocce, M., Gaztañaga, E. & Castander, F. J. (2015) The MICE grand challenge lightcone simulation–I. dark matter clustering. *Monthly Notices of the Royal Astronomical Society,* 448 (4), 2987-3000.

Github (2022) Machine learning methods for astrophysics. Available at: https://github.com/IftachSadeh/ANNZ (Accessed: 4th August 2022).

Goodfellow, I., Bengio, Y. & Courville, A. (2016) *Deep learning* MIT press.

Hildebrandt, H., van den Busch, J. L, Wright, A. H., Blake, C., Joachimi, B., Kuijken, K., Tröster, T., Asgari, M., Bilicki, M., de Jong, J. T. A, Dvornik, A., Erben, T., Getman, F., Giblin, B., Heymans, C., Kannawadi, A., Lin, C. -. & Shan, H. -. (2021) KiDS-1000 catalogue: Redshift distributions and their calibration. *Astronomy and Astrophysics (Berlin),* 647 A124.

IBM (2022) Neural Networks. Available at: https://www.ibm.com/uk-en/cloud/learn/neural-networks (Accessed: 11th July 2022).

Isobe, T., Feigelson, E. D., Akritas, M. G. & Babu, G. J. (1990) Linear regression in astronomy. *The Astrophysical Journal,* 364 104-113.

Kaggle (2022) Random Forest Methods, Differences and Real-life applications. Available at: https://www.kaggle.com/general/208443 (Accessed: 20th July 2022).

Kügler, S. D., Polsterer, K. & Hoecker, M. (2015) Determining spectroscopic redshifts by using k nearest neighbor regression. *Astronomy and Astrophysics (Berlin),* 576 A132.

Kuijken, K., Heymans, C., Hildebrandt, H., Nakajima, R., Erben, T., de Jong, Jelte T. A, Viola, M., Choi, A., Hoekstra, H., Miller, L., van Uitert, E., Amon, A., Blake, C., Brouwer, M., Buddendiek, A., Conti, I. F., Eriksen, M., Grado, A., Harnois-Déraps, J., Helmich, E., Herbonnet, R., Irisarri, N., Kitching, T., Klaes, D., La Barbera, F., Napolitano, N., Radovich, M., Schneider, P., Sifón, C., Sikkema, G., Simon, P., Tudorica, A., Valentijn, E., Verdoes Kleijn, G. & van Waerbeke, L. (2015) Gravitational lensing analysis of the kilo-degree survey. *Monthly Notices of the Royal Astronomical Society,* 454 (4), 3500-3532.

Mandelbaum, R., Seljak, U., Hirata, C.M., et al. (2008). *Monthly Notices of the Royal Astronomical Society* 386, (p. 781).

Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M. & Kerzendorf, W. E. (2013) Astropy: A community python package for astronomy. *Astronomy & Astrophysics,* 558 A33.

Sadeh, I., Abdalla, F. B. & Lahav, O. (2016) ANNz2: Photometric redshift and probability distribution function estimation using machine learning. *Publications of the Astronomical Society of the Pacific*, 128 (968), 104502.

TDS (2022) Machine Learning Basics with the K-Nearest Neighbours Algorithm. Available at: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 (Accessed: 10th July 20222).

Weinberg, D. H., Mortonson, M. J., Eisenstein, D. J., Hirata, C., Riess, A. G. & Rozo, E. (2013) Observational probes of cosmic acceleration. *Physics Reports*, 530 (2), 87-255.

Wright, A. H., Hildebrandt, H., van den Busch, J. L. & Heymans, C. (2020) Photometric redshift calibration with self-organising maps. *Astronomy and Astrophysics (Berlin)*, 637.