Towards Efficient Prediction of Neutral Hydrogen (HI) in Galaxies: An Automated Machine Learning Approach

Chaipat Tirapongprasert
Columbia University
(Dated: October 27, 2024)

HI content is pivotal in determining a galaxy's potential for future star formation and helps trace the distribution of gas in the universe. Since measuring HI directly (typically through radio telescopes like Arecibo) is time-consuming and limited to certain survey sizes, there is a strong need for alternative methods to estimate HI content. By developing an accurate predictive model, astronomers can leverage commonly available data, such as optical and morphological features, to approximate HI content more efficiently across large galaxy catalogs. This project thus aims to create a model that serves as a foundation for further research, automating machine learning processes to capture complex relationships among galactic features and HI content.

I. INTRODUCTION

The motivation for this project lies in the fundamental importance of understanding neutral hydrogen (HI) content in galaxies, since HI plays a critical role in galaxy formation, star formation processes, and overall galactic evolution. Based on established domain knowledge, we select 6 of the 41 features from the xGASS dataset, which are divided into two categories:

- magnitude-related properties: $model\_r$ (r-band model magnitude) and $NUVr$ (NUV-to-r color index)
- morphological properties: $expAB\_r$ (Axial ratio), $petrR50\_r$ (Radius enclosing 50% of Petrosian flux), $CINDX$ (Concentration index), and $INCL$ (Inclination angle)

Together, they form a robust input set for modeling the target variable, $lgGF$, or the logarithmic HI gas-to-stellar mass fraction. We now conduct multi-step exploratory data analysis (EDA) to examine relationships or potential predictive patterns between selected features and $lgGF$.

A. Feature Interrelationships

In the scatter plot between $INCL$ and $expAB\_r$, there is a tight clustering or alignment of points along a specific curve. Such predictable relationships can sometimes be redundant in modeling, as one feature could be directly derived from the other. This will be handled later in the model development stage.

It is worth noticing that features showing a consistent color gradient from one side to another (e.g., transitioning from red to blue or vice versa) indicate that as one feature value increases or decreases, the gas fraction tends to vary predictably. Such is observed in the scatter plot between $NUVr$ and $lgGF$.

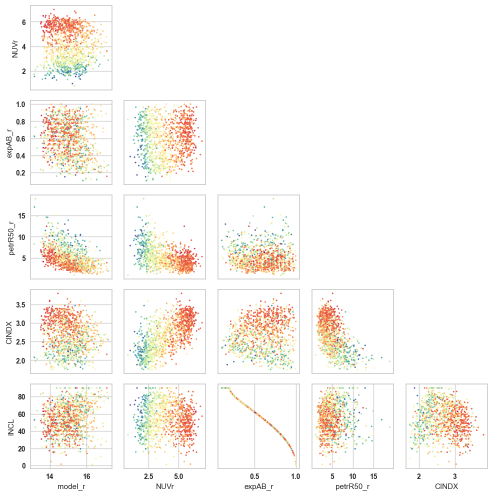

FIG. 1. Intercorrelation between variables

B. Box Plots for Gas Fraction

Since $NUVr$ seems to be strongly correlated with the gas fraction, let's bin $NUVr$ into intervals and create box plots for the gas fraction within each bin. This will show how gas fraction varies across different color ranges. Our figure below reveals a clear trend: as the NUV-r color index increases, there is a systematic decrease in the median log gas fraction, which aligns with our domain knowledge. Galaxies with lower (bluer) NUV-r values, typically associated with younger stellar populations and active star formation, show a higher median gas fraction. Conversely, galaxies with higher (redder) NUV-r values, often indicative of older, more evolved stellar populations, exhibit lower gas fractions.

C. Baseline Model

As always, standard preprocessing techniques, such as scaling and handling missing data, are applied to ensure consistency. We then apply a multivariate linear regres-

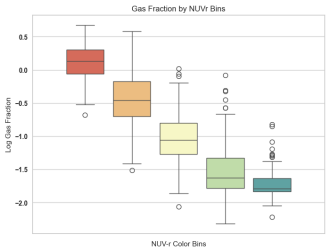

FIG. 2. Box plot

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	0.2159	0.0875	0.2953	0.8098	0.1543	0.9897	0.2420
gbr	Gradient Boosting Regressor	0.2155	0.0882	0.2959	0.8084	0.1540	0.9232	0.0380
rf	Random Forest Regressor	0.2183	0.0890	0.2973	0.8066	0.1572	0.9741	0.0590
huber	Huber Regressor	0.2233	0.0895	0.2982	0.8047	0.1526	0.8594	0.0090
et	Extra Trees Regressor	0.2224	0.0899	0.2990	0.8044	0.1576	0.9477	0.0430
lr	Linear Regression	0.2256	0.0904	0.2999	0.8026	0.1539	0.8895	0.0310
ridge	Ridge Regression	0.2258	0.0905	0.2999	0.8026	0.1538	0.8894	0.0120
lar	Least Angle Regression	0.2256	0.0904	0.2999	0.8026	0.1539	0.8895	0.0090
br	Bayesian Ridge	0.2258	0.0905	0.3000	0.8026	0.1538	0.8894	0.0090
lightgbm	Light Gradient Boosting Machine	0.2282	0.0974	0.3113	0.7884	0.1619	1.0305	0.2670
xgboost	Extreme Gradient Boosting	0.2337	0.0998	0.3150	0.7834	0.1638	1.0045	0.0310
omp	Orthogonal Matching Pursuit	0.2517	0.1082	0.3278	0.7649	0.1652	1.0722	0.0100
knn	K Neighbors Regressor	0.2558	0.1164	0.3402	0.7488	0.1737	1.1049	0.0120
ada	AdaBoost Regressor	0.2850	0.1184	0.3435	0.7424	0.1712	1.1087	0.0230
dt	Decision Tree Regressor	0.3000	0.1692	0.4103	0.6315	0.2107	1.3755	0.0120
par	Passive Aggressive Regressor	0.3635	0.2141	0.4547	0.5421	0.2206	1.5910	0.0090
en	Elastic Net	0.4781	0.3223	0.5667	0.3045	0.2716	3.2663	0.0090
lasso	Lasso Regression	0.5836	0.4708	0.6850	-0.0163	0.3269	3.9524	0.0100
llar	Lasso Least Angle Regression	0.5836	0.4708	0.6850	-0.0163	0.3269	3.9524	0.0090
dummy	Dummy Regressor	0.5836	0.4708	0.6850	-0.0163	0.3269	3.9524	0.0100

FIG. 3. Comparing all model

sion as our baseline model. The model employs the normal equation to estimate the optimal coefficients, while cross-validation is performed to obtain a robust assessment of model performance. Typically, in k-fold cross-validation, the data is divided into k parts, with each fold used once as a test set while the others serve as training data. This process is repeated k times, providing an average performance measure and ensuring that the model will generalize to new data and not overfit or underfit. It is, however, a little inconvenient to have to manually set up cross-validation each time, which motivates us to come up with a fresh strategy that will be covered in the main methodology section.

The baseline linear regression yields an average R^2 of approximately 0.768 across five cross-validation folds, suggesting that the selected features explain about 76.8% of the variance in $lgGF$. This implies a moderate predictive power at best, but we can do better. Note that even with the straightforwardness of such a linear model, implementing multivariate regression with cross-validation, parameter tuning, and feature scaling is somewhat tedious. Therefore, we opted to simplify the workflow using automated machine learning, facilitated by PyCaret Regressor. This approach significantly helps with identifying top-performing models and highlighting feature importance.

II. MAIN METHODOLOGY

A. Model Selection

First, PyCaret's setup function initializes the regression environment, standardizing the dataset for machine learning. Our modeling setup incorporated six predictor features. This initial setup performed essential preprocessing steps, such as imputing missing values and scaling continuous features, ensuring data consistency and feature comparability across models. With good grace, our data is already relatively clean manually in the last section.

B. Model Development

We then run every regression models using the *compare_models*() function to find that CatBoost Regressor, Gradient Boosting Regressor, and Random Forest Regressor perform best. It is not surprising that Catboost ranks first given its unique handling of feature interactions, bias reduction techniques, and computational efficiency.

By using PyCaret's *create_model* function, each model was initialized with default hyperparameters to benchmark initial performance. The PyCaret framework also provided immediate cross-validation metrics for each model, facilitating a quick comparison of baseline performance across the selected algorithms.

C. Tuning and Blending

To refine each model, we employed PyCaret's *tune_model* function, which optimizes hyperparameters through grid search, enhancing each model's performance without manual tuning. Parameters such as the learning rate, maximum tree depth, and the number of estimators were systematically adjusted, improving the accuracy of each model. We also did feature importance analysis using $plot\_model(best\_model, plot = \text{“}feature\text{”})$, which revealed the NUV-to-r color index (NUVr) as a dominant predictor within each tuned model, underscoring its critical role in HI prediction as laid out in the EDA process.

Ensembling techniques were then applied to each tuned model to improve stability and predictive accuracy. Here, we use *blend_models* to average the predictions of multiple models to stabilize performance, providing a balanced, low-variance approach.

D. Finalizing

Lastly, to assess final model accuracy on unseen data, we finalized the best-performing model using $finalize\_model$. This function locks the model parame-

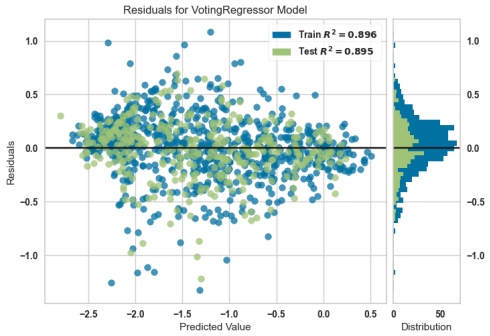

FIG. 4. Residual Plot

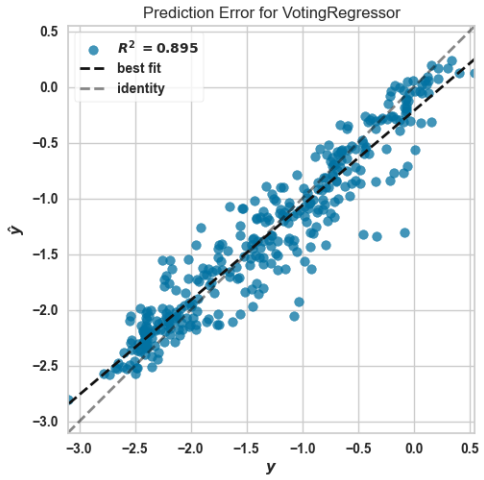

FIG. 5. Prediction Error Plot

ters and retrains the entire training set, preparing it for deployment on the hold-out sample. Predictions on the hold-out set provide a realistic assessment of model generalization.

III. CONCLUSIONS

In this project, we have automated the machine learning pipeline and combined the outputs from many models, resulting in significant improvements in predictive accuracy. Through a comparison of individual and combined approaches, we found that blending the ensembled CatBoost, GBR, and RF models achieved the best performance, outperforming each model on its own. Our blended model leveraged the strengths of each base learner, providing enhanced stability and accuracy, as indicated by increased R^2 scores ($0.809 \rightarrow 0.895$) and reduced error on the hold-out sample ($0.2953 \rightarrow 0.2015$). The blended model's superior performance demonstrates a valuable approach for stabilizing predictions across diverse galaxy features and capturing complex relationships in the data. It also highlights the utility of PyCaret's ensembling and blending functionality for efficient model refinement. With its low-code approach, PyCaret facilitated rapid experimentation with various algorithms and configurations, allowing us to determine an optimal baseline model that can be readily adapted or expanded upon in future research.

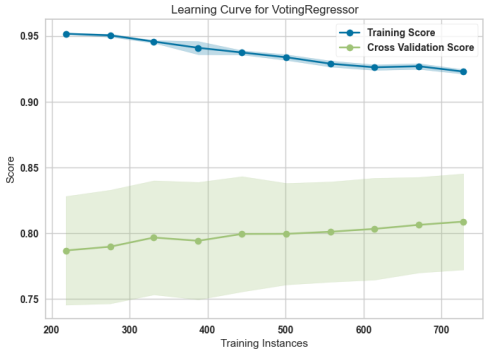

FIG. 6. Learning Curve