Stellar Spectral Classification with Active Learning Approach

[1,a]Kishan Malaviya, [1,b]Riya Mahendra, [1,c]Yogesh Patel
[1]Mathematical and Physical Sciences division, School of Arts and Sciences, Ahmedabad University, India - 380009
[a]kishan.m1@ahduni.edu.in, [b]riya.m5@ahduni.edu.in, [c] yogeshkumar.p@ahduni.edu.in

***Abstract*—Stars are crucial part of any stellar system. The distribution of different types of stars provides information of the evolution of the stellar system where they are found. Therefore, accurately identifying the star type is of great importance in astronomy. The task of classification of star can be handled using machine learning techniques. Here, we present a comprehensive study of performance of various machine learning techniques for classification of stars. We find that Active learning approach performs better than the traditional machine learning algorithms in this context.**

Keywords : SDSS - Active Learning - MaStar - Spectral classification - Machine Learning

I. Introduction

Study of stars is central in astronomy, particularly due to their important role in galactic dynamics and galaxy evolution. Stars are classified primarily using the observed spectra of them. Different chemical composition and different physical properties of stars yields variety of features in stellar spectrum. Therefore, one can use the spectral features in a star's spectra to identify its type. The most widely known and simplest classification scheme for stars is the Harvard Scheme where stars are classified in classes : O, B, A, F, G, K, M. This sequence is defined on the basis of the surface temperature of stars (T_{eff}) where from O ($T_{eff} \gtrsim 25000K$) to M ($2000K < T_{eff} < 3500K$) the surface temperature decreases (see [1] for a detailed review).

Before the advent of machine learning (ML) techniques, stellar classification was done by visual inspection of spectral features by human experts. Currently, owing to the huge abundance of data from various telescopes, the manual classification of stars is not feasible. Many studies have performed the spectral classification using different ML techniques (e.g. [2] and reference therein). However, there are limitations of machine learning methods, one of these limitations is : need for the labeled data for training a ML model. With the availability of huge amount of data from different surveys and telescopes, it is difficult and time consuming to create large labelled datasets to train models for better performance. The active learning approach of machine learning is effective in achieving better accuracy in such cases where the amount of labelled data is less for training the ML model [2].

In this work, we quantitatively evaluate the performance of different ML techniques along with the active learning approach specifically in the context of stellar classification problem. We consider three different classification of stars which are based on :

- T_{eff} (Surface Temperature) - classes : O, B, A, F, G, K, M
- $log(g)$ (Surface gravity) - classes : C1, C2, C3, C4, C5, C6
- $[Fe/H]$ (metallicity) - classes : XMR, MR, MP, XMP

We use the feature set to perform classification based on above mentioned classes and evaluate different performance metric for them to draw comparisons. In the following sections, we mention the data, methodology and results obtained from our study. At last, we list the key finding of the work.

II. Data

In this study, we used the MaStar library from SDSS (Sloan Digital Sky Surveys) DR17. It is a large and higquality empirical stellar library containing real (observed) stellar spectra, commonly used for calibration and research. The MaStar library covers a wide range of spectra whose wavelength ranges from 3622 $\mathring{A}$ to 10,354 $\mathring{A}$, having a spectral resolution of $R \sim 1800$. The dataset contains total of 59,805 spectra with more than 85% of spectra having SNR $>$ 50 of 24,162 unique stars. This data covers a wide range of physical parameters described as follows :

- 2800 K $\lesssim T_{eff} \lesssim$ 31000 K
- - 0.25 dex $\lesssim$ log(g) $\lesssim$ 5.25 dex
- -2.75 dex $\lesssim$ [Fe/H] $\lesssim$ 1.00 dex

To prepare the data for training ML models, we followed the steps prescribed in [2]. These steps are briefly described in the following section.

III. Methodology

In our study, dataset includes 4563 features, which makes the dataset very huge. So, here we have to reduce the number of features such that the variance need to be maximum. We employed the data pre-processing steps prescribed in [3], 2025. Initially, we reduced the features from 4563 to 170. This is done by keeping the flux values at wavelength values around certain spectral emission lines. This is done because different spectral types are associated with different characteristic emission lines in the spectra. For example, Ca II K (3934 Å) and H (3968 Å) lines are important markers in A-type stars, whose intensity depends on temperature and luminosity effects [4]. Likewise, the Fe I (4046 Å) line, usually employed in combination with the Hδ hydrogen line, is important in temperature classification, especially in F-type and later stars [4]. For B-to-M-type star classification, two spectral lines are important: H δ (4102 Å), found in A, B, F, and G-type stars, and Ca I (4227 Å), found in F, G, K, and M stars [5]. The G-band CH (4300 Å) feature is an important feature, which is prominent in late-G to K-type stars and sensitive to surface gravity changes [4]. In O-type stars, the He II (4686 Å) line prevails [4], whereas the TiO band (4955 Å) is an important marker for M-type star classification, with at least one TiO band [6]. Finally, the Fe I (5269 Å) and Fe II (5018 Å) lines are important in metallicity classification [7] . So we dropped other wavelength which makes our data lighter. and we will have the 170 different wavelengths left out of 4563. Further, we apply the PCA (principle component analysis) to the 170 components, and we get the 99.95% variance in 9 PCA components. Hence, finally we are left with 9 features which will be used for classification task.

We want to classify this data on 3 different bases: 1) classification with respect to the temperature of the stars, surface gravity (Log g) of stars, and the metallicity of the stars (Fe/H). This dataset already consists of features inside the data like metallicity, effective temperature, surface gravity,and associated errors with them. It also includes the classification based on effective temperature, surface gravity, and metallicity. In this set of spectra, more than 85% have signal-to-noise ratios (SNR) that increase the trustworthiness of the data.All of this data is managed using the manga ID.

For data filtration, we used a threshold based on VAC(Value added catalog) (only included those stars that have been included in VAC) and the signal-to-noise ratio (SNR). After dropping the data, we got the 59085 species of 24162 different stars. For these spectra, we have $SNR > 50$ with an overall mean of 126.

So after data processing we are having 59085 rows with 9 features. Now, once the data was prepared using the above mentioned procedure, we employed different machine learning techniques. The results from them are discussed in the next section.

IV. Results and discussion

Here, we present and discuss the results of the experiments carried out in this study.

A. K-Nearest Neighbors and SMOTE

We begin by implementing the unweighted and weighted K-Nearest Neighbors (K-NN) algorithm. The weighted K-NN yields higher accuracy as it prioritizes closer neighbors by assigning higher weights based on proximity. The performance of the classifier was evaluated based on confusion matrices. It reveals that the classifier performs well for dominant classes: K, G, F in T_{eff}; C5, C6 in $\log(g)$; and MR, MP in [Fe/H]. However, misclassification is observed between adjacent classes due to their similar spectral characteristics.

While weighted K-NN mitigates some effects by emphasizing closer neighbors, class imbalance remains a critical factor influencing the model's performance. To address this, we implemented SMOTE (Synthetic Monitoring Oversampling Technique) on the training data and re-evaluated the K-NN performance. Metrics used include precision, recall, F1-score, and accuracy.

We compared the results before and after applying SMOTE and found that before SMOTE, high precision and recall for dominant classes and very low F1-score for minority classes. However, after SMOTE, results increase in overall accuracy with significantly improved F1-scores for underrepresented classes for all three stellar parameters (Table I).

Macro F1 computes the F1-score independently for each class and then takes the average by treating all classes equally, regardless of the number of samples and is useful for understanding model performance on minority classes. Meanwhile, the Weighted F1 score computes the average weighted by the number of true instances (support) in each class, which indicates the model's overall performance, especially when the dataset is imbalanced.

TABLE I
Performance of K-NN Classifier Before and After Applying SMOTE

Parameter	Metrics	Before SMOTE	After SMOTE
T_{eff}	Accuracy	84%	88%
	Macro F1	0.60	0.81
	Weighted F1	0.83	0.88
Log(g)	Accuracy	69%	70%
	Macro F1	0.47	0.67
	Weighted F1	0.65	0.71
[Fe/H]	Accuracy	84%	85%
	Macro F1	0.49	0.75
	Weighted F1	0.81	0.87

B. Ensemble Learning Approach

In this study, ensemble learning was used to enhance the performance of our classifier by leveraging the strengths of multiple models, because individual classifiers often suffer from overfitting or underfitting, especially with imbalanced datasets. Here we used two ensemble models: Random Forest (ensemble of decision trees) and Voting Classifier (ensemble of different types of models). Here, a soft voting classifier has been used that combines the predictions of all three base learners: K-Nearest Neighbors (K-NN), Random Forest (RF), and Histogram-based Gradient Boosting (GB) and trained on resampled (balanced) training data. The final prediction was made by averaging the predicted probabilities from all the models.

In this, models have assigned different weights (2,2,1) for Gb, RF, and K-NN, respectively. This will reflect their performance contributions. The metrics used for the performance evaluation are: AUC, MCC, Sensitivity, Specificity. Area under the Curve (AUC) helps in distinguishing between positive and negative classes for different classifiers. Matthews Correlation

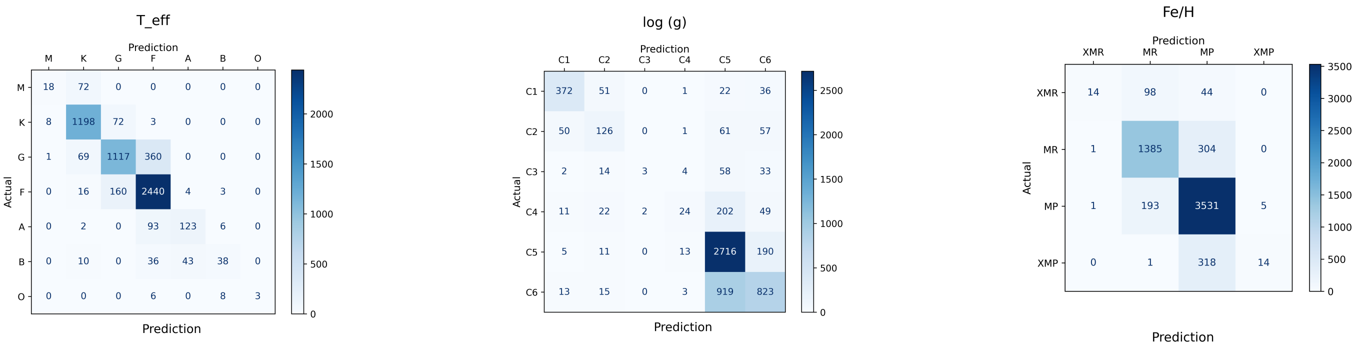

Fig. 1. Confusion matrices for K-NN classifier predictions on T_{eff}, log(g), and [Fe/H].

Coefficient (MCC) provides a balanced view of classification quality, by looking at both the correct and incorrect predictions for both classes. Sensitivity tells us how well a model identifies actual positive instances, whereas specificity indicates how well a model identifies negative instances. The results for all three parameters are shown in Table II.

TABLE II
PERFORMANCE METRICS FOR ENSEMBLE MODELS

Parameter	Model	AUC	MCC	Sensitivity	Specificity
T_{eff}	KNN	0.966	0.834	0.895	0.977
	RF	0.990	0.899	0.888	0.986
	GB	0.992	0.885	0.901	0.984
	Voting	0.995	0.897	0.909	0.986
log(g)	KNN	0.912	0.589	0.761	0.932
	RF	0.956	0.670	0.768	0.944
	GB	0.937	0.576	0.715	0.930
	Voting	0.958	0.651	0.776	0.941
[Fe/H]	KNN	0.956	0.781	0.878	0.955
	RF	0.983	0.844	0.862	0.965
	GB	0.981	0.774	0.879	0.954
	Voting	0.986	0.822	0.885	0.963

C. Active Learning Approach

In Active Learning, we divided our total data (59805 spectra) in three datasets:

1) Training set – 200 spectra
2) Pool of unlabelled data – 47,108 spectra
3) Test set – 11,777 spectra

In this study, we have implemented uncertainty sampling for querying the data from the pool of unlabelled data. In uncertainty sampling, the model randomly selects a new sample from the pool of unlabelled data and then asks the oracle for the label corresponding to it. Here, we used Random Forest Classifier as our base model and performed Active learning for different number of queries. To draw a comparison between traditional ML and Active learning, we compare the accuracy score obtained from only Random Forest model (i.e. traditional ML) and Random Forest + Active Learning for classification of stars in terms of T_{eff}.

In traditional ML with a Random Forest classifier, we allocated 80% data as the training set and 20% as a test set. With this approach, we gained an 82.5% accuracy score. In Active Learning with Random Forest classifier, we allocated 200 samples (out of 59805) as the training set and out of the rest dataset, we allocated 20% as a test dataset and the rest as unlabelled pool dataset. With this approach, we gained 80% accuracy score with just 250 queries.

TABLE III
ACTIVE LEARNING PERFORMANCE FOR VARYING QUERIES

No. of Queries	Unlabelled Pool Size	Accuracy Score (%)
0	47108	76.54
50	47058	78.52
100	47008	78.90
150	46958	79.84
200	46908	79.77
250	46858	80.77

Hence, we show that with the limited dataset, the Active Learning approach gives better results compared to traditional ML. We show the change in the accuracy score with the number of queries in Table III, for illustrating the Active Learning approach quantitatively.

V. CONCLUSION

In this work, we evaluated the performance of different machine learning techniques for classification of stars in terms of : (1) surface temperature (T_{eff}), (2) surface gravity ($log(g)$), (3) metallicity ([Fe/H]). We employed MaStar library from SDSS DR17 to train and test our models. Based on our analysis we conclude the following :

- The data from MaStar - SDSS DR17 is highly imbalanced. To handle the imbalance we employed stratified sampling and SMOTE. We find that better accuracy is achieved in the case of SMOTE.
- Ensemble classifier is seen to be providing better results than individual models. With this approach, we could achieve upto 98.6%, 94.1% and 96.3% for classification by T_{eff}, $log(g)$ and [Fe/H] respectively.
- Using active learning with uncertainty query sampling method, we found that we could achieve better accuracy with very few number of samples. We showed that, with traditional random forest method, we achived 82.5%

accuracy with 80/20 train/test split. On the other hand, with active learning, we could reach 80% accuracy using only 200 initial samples and 250 queries.

Acknowledgment

The authors would like to thank Professor Mehul Raval, Ahmedabad University for his support and encouragement throughout this project.

References

[1] S. Giridhar, "Spectral classification: Old and contemporary," in *Principles and Perspectives in Cosmochemistry: Lecture Notes of the Kodai School on'Synthesis of Elements in Stars' held at Kodaikanal Observatory, India, April 29-May 13, 2008*, pp. 165–180, Springer, 2010.

[2] R. El-Kholy and Z. Hayman, "Optimised sampling of sdss-iv mastar spectra for stellar classification using supervised models," *Astronomy & Astrophysics*, vol. 693, p. A300, 2025.

[3] R. El-Kholy and Z. Hayman, "Optimised sampling of sdss-iv mastar spectra for stellar classification using supervised models," *Astronomy and Astrophysics*, 2025.

[4] R. O. Gray, C. J. Corbally, and A. J. Burgasser, *Stellar Spectral Classification*. Princeton Series in Astrophysics, Princeton: Princeton University Press, 2009.

[5] M. J. Brice and R. Andonie, "Aj," *Astronomical Journal*, vol. 158, p. 188, 2019.

[6] R. O. Gray and C. J. Corbally, "Aj," *Astronomical Journal*, vol. 147, p. 80, 2014.

[7] N. C. Santos, G. Israelian, and M. Mayor, "Spectroscopic [fe/h] for 98 extra-solar planet-host stars," *Astronomy and Astrophysics*, vol. 415, no. 3, pp. 1153–1166, 2004.