Variable Star Classification

Nikash Prakash

December 12, 2022

Abstract

A reproduction of Comparing Multiclass, Binary, and Hierarchical Machine Learning Classification schemes for variable stars. The motivation behind this reproduction is to validate that their method and classification schema has it's claimed accuracy rate.

1 Introduction

The goal of this project is to classify variable stars. Variable stars have three broad classes: eclipsing, pulsating, and rotational. Eclipsing binary variable stars have further sub-types: detached binaries (EA) and contact/semidetached binaries (Ecl). Rotating variable stars break down into: ellipsoidal variables (ELL) and spotted. But since rotational variables typically have low amplitudes (< 0.2 apparent magnitude), no clear distinction can be made.[DDC+17] As such a combined rotational class (ROT) is used. Pulsating variables' sub-types: RR Lyrae, Cepheids, long-period variables (LPVs), and δ-Scutis. These sub-types can be further classified, RR Lyrae: RRab (fundamental mode of the Fourier spectra), RRc (first overtone), RRd (multimode), and Blazhko; Cepheid: anomalous Cepheids (ACEPs) classical Cepheids (Cep-I), Type II Cepheids (Cep-II).

Astronomy has evolved in the last two decades to produce massive amounts of data, too much for people to interpret manually. For example the Sloan Digital Sky Survey has over 230 million celestial objects, and the Large Synoptic Survey Telescope (LSST) produced around $\sim$15 terabyte (TB) of raw data per night. Classification of new objects will further observational astronomy and our general understanding of the universe.

In this project, 3 types of classification: multi-class, binary, and hierarchical schemes will be used to classify variable stars. Statistical features are extracted for each star, and run classification using those features. Next is to calculate performance measures (i.e. Precision, Recall/Sensitivity, Specificity, etc.) to compare the schemes. The method of hierarchical classification led to good results allowing for identification of sub-types of Cepheids and eclipsing binaries with a balanced-accuracy rate of 81% and 86% respectively [HLSM19].

2 Related work

There is some research regarding the classification of variable stars. For example, the semi-supervised hierarchical method + clustering analysis[PCPP22] and a package for automated classification [KBJ16]. The semi-supervised hierarchical learning achieved 90% accuracy only using 5% of the data-set to train the model. And clustering analysis confirmed that most clusters had purity over 90% with respect to classes and 80% with respect to sub-classes. This is a strong method for classification of variable stars as it doesn't require an extensive training set. Their method is supported by dimensionality reduction of the data to avoid the curse of dimensionality when modeling and for visualization. The UPSILON package analyzed the light curves to classify them, using 16 feature variables, such as A which ratio of the squared sum of residuals of magnitudes that are either brighter than or fainter than the mean magnitude. They are somewhat similar as they applied multi-class classification with a random forest. The achieved good results when a class had enough data points (¿0.85 precision). They has a standard approach of analyzing the physical features of the stars (i.e. apparent magnitude over time). Both studies had state of the art modeling, they processed tens to hundreds of thousands of stars. The methods used here [PCPP22], were clever as they saw a deficiency in the data set (lack of labels) and found another method to classify the stars.

3 Data and Pre-Processing

The CRTS project has released their data - Catalina Surveys Data Release 2 (CSDR2) - the study focuses on the Catalina Surveys Southern Sky Periodic Variable Catalog (SSS PV Catalog).

3.1 Data Cleaning

Due to there being an imbalance in the data set, significantly more Ecl variables (type 5) than others, down-sampling from 18803 to 4509 (number of EAs) is necessary. The study removes LMC-Cep-I (type 13) because only 10 samples exist and these objects are similar to ACEPs. LMC-Cep-I are classical Cepheids in the Large Magellic Cloud, one of the Milky Way's satillite galaxies. It finally removes the miscellaneous variable stars (type 11) due to difficulty in classification [HLSM19]. At this point we are left with $23,143$ stars.

Next was to remove outliers in the data. Removed any variable star that had a apparent magnitude (how bright it looks to the telescope) greater than 3 standard deviations from the mean, for a total of 17 stars removed. See Figure 3a

3.2 Feature Extraction

The study does not use normal features such as apparent magnitude, amplitude, etc. but instead statistical features since they have "no preconceived notions of their suitability or expressiveness as input feature to the algorithms" (**(author?)**). Such features include location (mean magnitude), scale (standard deviation), variability (mean variance), morphology (skew, kurtosis, and amplitude), and time (period). The amplitude in morphology is half the difference between the median of the maximum 5% and median of the minimum 5% magnitudes. The next step is normalizing these features, a feature X_i^j is divided by $\max X_i^j$, leading to each feature re-scaled.

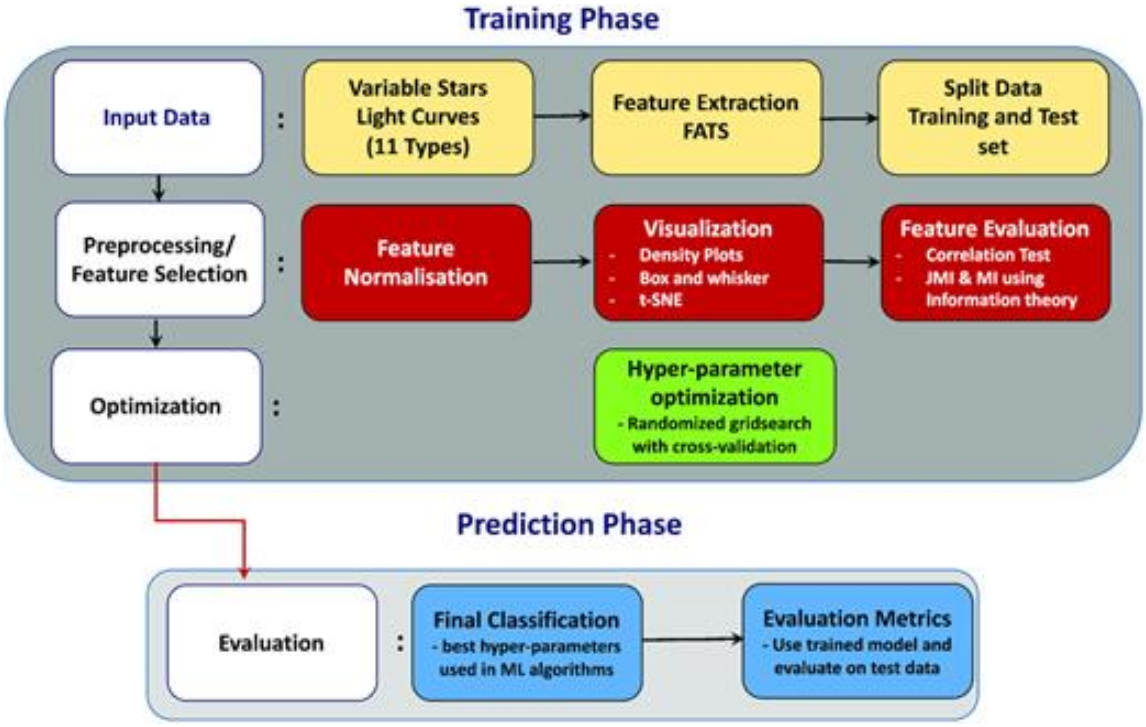
$$X_{Train} = |X_{Train_i} \max(X_{Train_i})|, \quad X_{Test} = |X_{Test_i} \max(X_{Train_i})|$$

3.3 Pre-Processing

The data-set is split into 70% training and 30% testing/validation. In order get the data ready for visualization, split training and testing data by class sample size his is done to ensure that then filter by classes to remove classes that were not suitable for visualization (ex: '3', '2','5'). Binary classification requires there to only be 2 levels, one example for this is the training set only with star types 1 and 6 (RRabs, EAs).

4 Methods

Figure 1: Throughout the body of research, this classification pipeline is followed



4.1 Algorithms

For each multi-class, binary, and hierarchical classification we use the three following algorithms KNN, Decision Trees, and Random Forests.

1. **K-Nearest-Neighbors** is a distance base algorithm, it is an instance-based algorithm that assigns unlabelled points based of the label of its k-nearest neighbours. KNN has good performance when the size training data set is large. But K-Nearest-Neighbours suffers from the curse of dimensionality and bias. Since it requires all features, any small set of outliers/irrelevant data and the distance between instances will biased by the irrelevant samples and their values.

2. **Decision Trees** are a tree-based method that attempts to split data recursively according to the data's feature values, each split creates a new branch. Each branch will end at a leaf node with a class/label. The goal is to build a tree that can guide decisions that accurately separates examples traveling down the tree until it reached the correct leaf node. Generally, singular decision trees for classification have poor performance due to a low variance, high bias towards the data-set. Which would lead the decision tree to overfit to the training data, and have low accuracy or low scoring results for a performance measure on the testing set. A change in the data would change the structure of the classification tree.

3. **Random Forests** help to solve the problem that decision trees face, due to the face that they consider $m \approx \sqrt{p}$ predictors where p is the total number of predictors. At every split in the tree is has to consider m random predictors, that decorrelates the trees which in turn reduces the variance because it averages uncorrelated quantities. This suggests that random forests will not overfit if the number of trees B increases, however if said trees are closely correlated then the variance reduction effect will grow limited.

4. **K-fold cross validation** (k-fold CV) is a method used to reduce over/underfittng which can happen if training and evaluation happen on the same data. The way K-fold CV works is by randomly splitting the training set into K different folds. The classification algorithm is trained and tested K times, it uses $K-1$ folds to train and the remaining fold is used to validate the model, repeat this process until you have performed validation on every fold.

5. **t-SNE** is a method that allows high-dimensional objects to have their dimensions reduced and re-presentable in 2 or 3 spacial dimensions. It works by clustering similar high-dimensional objects with nearby points using k-dimension tree (k-d tree) of all the points. A t/Cauchy-distribution is used to calculate the euclidean distance between each point and its kNN. This distance is then mapped to a probability distribution. Similar points have high probability of being assigned to the same class and different points the opposite. Afterwards, t-SNE creates a new probability distribution using a gradient descent method, in low-dimensional space over the points, in turn minimizing the Kullback-Leibler divergence between the two distributions. The visualization can be seen in Figure 3b.

4.2 Information theory

The point biserial correlation, r_{pb} is applied to find the relationship between a continous variable, x, and binary variable, y. It is on the range

Features that may appear informational poor could provide new meaningful information when analyzed with one or more other features. Information theory is generally based on the entropy of a feature X. To start measuring the importance of the features the mutual information - the amount of information between input feature X and true class label Y is defined as

$$I(X;Y) = H(X) - H(X|Y)$$

MI is zero when X and Y are statistically independent. Therefore, it is useful for features to have high MI. A high MI indicates that a feature is correlated with the target variable.

5 Results

The primary metrics used were Precision $= \frac{TP}{TP+TN}$, Recall$= \frac{TP}{TP+FN}$,
F1-score $= 2 * \frac{Precision * Recall}{Precision+Recall}$, Geometric mean score $= \sqrt{\frac{TN_i}{TN_i+FP_i} + \frac{TP_i}{TP_i+FN_i}}$, and Balanced Accuracy $= \frac{1}{|C|} \sum_{i \in C} \frac{TP_i}{TP_i+FN_i}$,
where i is a specific class in C, which is the set of all classes.

Feature	Mean	Std	Mean Variance	Skew	Kurtosis	Amplitude	Period
r_{pb}	-0.552171	-0.469722	-0.352047	0.583856	0.241811	-0.365366	0.393613
M.I.	0.335864	0.306272	0.267088	0.199604	0.134023	0.098962	0.087934
JMI	6	4	7	2	3	5	1

Table 1: The point biserial correlation coefficient and the mutual information MI(X; Y) for each feature for the binary class pair: RRab (Type 1) and EA (Type 6) are illustrated.

Classifiers	Precision	Recall	F1-score	G-mean	Balanced accuracy
Type 1 and 6 classification					
RF	0.951/0.966	0.951/0.966	0.951/0.966	0.959/0.959	0.959/0.959
means	∼0.959	∼0.959	∼0.959	0.959	0.959
DT	0.902/0.932	0.931/0.904	0.916/0.918	0.917/0.917	0.917/0.917
means	0.917	∼0.918	∼0.917	0.917	0.917
KNN	0.904/0.912	0.908/0.908	.906/0.91	0.908/0.908	0.908/0.908
means	0.908	0.908	0.908	0.908	0.908
Type 9 and 10 classification					
RF	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
means	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
DT	0.981/0.969	0.972/0.979	0.976/0.974	0.975/0.975	0.975/0.975
means	0.917	∼0.918	∼0.917	0.917	0.917
KNN	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
means	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0

Table 2: Two values are given for all metrics. For each binary case presented, the first values represent metric values for Type 1 (RRab) and Type 9 (δ Scuti). The second values are the metrics values for Type 6 (EA) and Type 10 (ACEP).

The hyperparameters used where selected using randomized grid search, and used stratified 5-fold cross-validation on the training set to evaluate model performance. The optimal parameters for Random Forest found are numTrees = 990, min-samples-split= 2, min-samples-leaf=1, max-features=square root, max-depth=90, criterion=entropy. To ensure that the model doesn't over-fit we use stratified 5-fold cross validation due to a large amount of unbalanced data.

The results of multiclass classification using an RF classifer can be seen in Figure 4a. The RF classifier got a balanced accuracy of : 0.5895, K-Nearest-Neighbors got a balanced accuracy of : 0.373, and Decision tree got a balanced accuracy of : 0.464. All of the algorithms does not perform well, this is expected given the large class imbalance (i.e. small sample sizes make up $\sim 6\%$ of all the data). Focusing on results from the RF classifier as it performed the best. We can see that most misclassified examples happen from low training examples indicating that there is bias toward large sample size classes.

Binary Classification did better than multiclass classification by breaking up the classes that are roughly balanced. To determine if two classes are separable we plot the box and whisker plots for the features extracted, we test Type 1 (RRab) as the negative class and Type 6(EA) as the positive class. In Figure 4b we can see that all the feature except the period show enough separability between Types 1 and 6. Period was used for classification because of its r_{pb} value and joint mutual index ranking see in Table 1. The three classifiers have been trained on two balanced pairwise combinations from the large sample and small sample data: Types 1 and 6, Types 9 (δ-Scuti) and 10 (ACEP). We observe that Random Forests and KNN perfectly classify Types 9 and 10 balanced accuracy 1.0 with G-mean value of 1.0, while in multiclass classification they had some misclassification. Results are summarized in Table 2

Hierarchial classification breaks down the problem into 3 layers, the first has the 3 general classes: Eclipsing, Pulsating, and Rotational. Which helps with some class imbalance, however since rotational class has many less examples this separation is not perfect. The results can be seen in Figure 2 and Table 3, the table is organized by layer, indicated by class level and index column in the table (i.e. first layer is index 0).

6 Conclusion

The random forest classifier under binary classification is the highest performing in my work, however in [HLSM19] they obtain random forest classifier under hierarchical classification as the highest performing method. This could

index	Precision	Recall	F1-score	G-mean	Balanced accuracy	Class l...
0	0.767/0.664/0.819	0.772/0.536/0.872	0.770/0.593/0.844	0.813/0.712/0.855	0.772/0.536/0.872	1
1	0.992/1.000/0.958/0.917	0.998/0.993/0.902/0.828	0.995/0.996/0.929/0.870	0.979/0.996/0.949/0.909	0.998/0.993/0.902/0.828	2
2	0.930/0.912	0.910/0.931	0.920/0.921	0.921/0.921	0.910/0.931	2
3	0.942/0.913/0.690/0.400	0.995/0.979/0.252/0.035	0.968/0.945/0.369/0.065	0.969/0.953/0.500/0.187	0.995/0.979/0.252/0.035	3
4	0.860/0.907	0.915/0.848	0.887/0.876	0.881/0.881	0.915/0.848	3

Figure 2: RF classification for the different hierarchical layers

Layer	Precision	Recall	F1-score	G-mean	Balanced accuracy
1	0.736	0.565	∼0.736	∼ 0.8	0.726
2:Pulsating	∼0.967	∼0.93	∼0.948	0.959	0.93
2:Eclipsing	∼0.921	∼0.921	∼0.921	∼ 0.921	∼0.921
3:RR Lyrae	∼0.736	∼0.565	∼0.586	∼0.739	0.565
3:Cepheids	∼0.883	∼0.881	∼ 0.882	0.881	0.881

Table 3: Means of each metric for each layer in hierarchical classification

be due to differences in how they separate the data and how I separate it, and their optimal hyperparameters are likely different as well. In the future, to improve this classification pipeline I would try other tree methods such as XGBClassifier (eXtreme Gradient Boosting a boosting algorithm based on gradient boosted decision trees algorithm), and unsupervised learning techniques such as convoluted-neural-networks. Unsupervised learning would make it applicable to data sets that haven't been manually classified, a more general model would a benefit due to the hundreds of thousands of stars that have not been process yet.

References

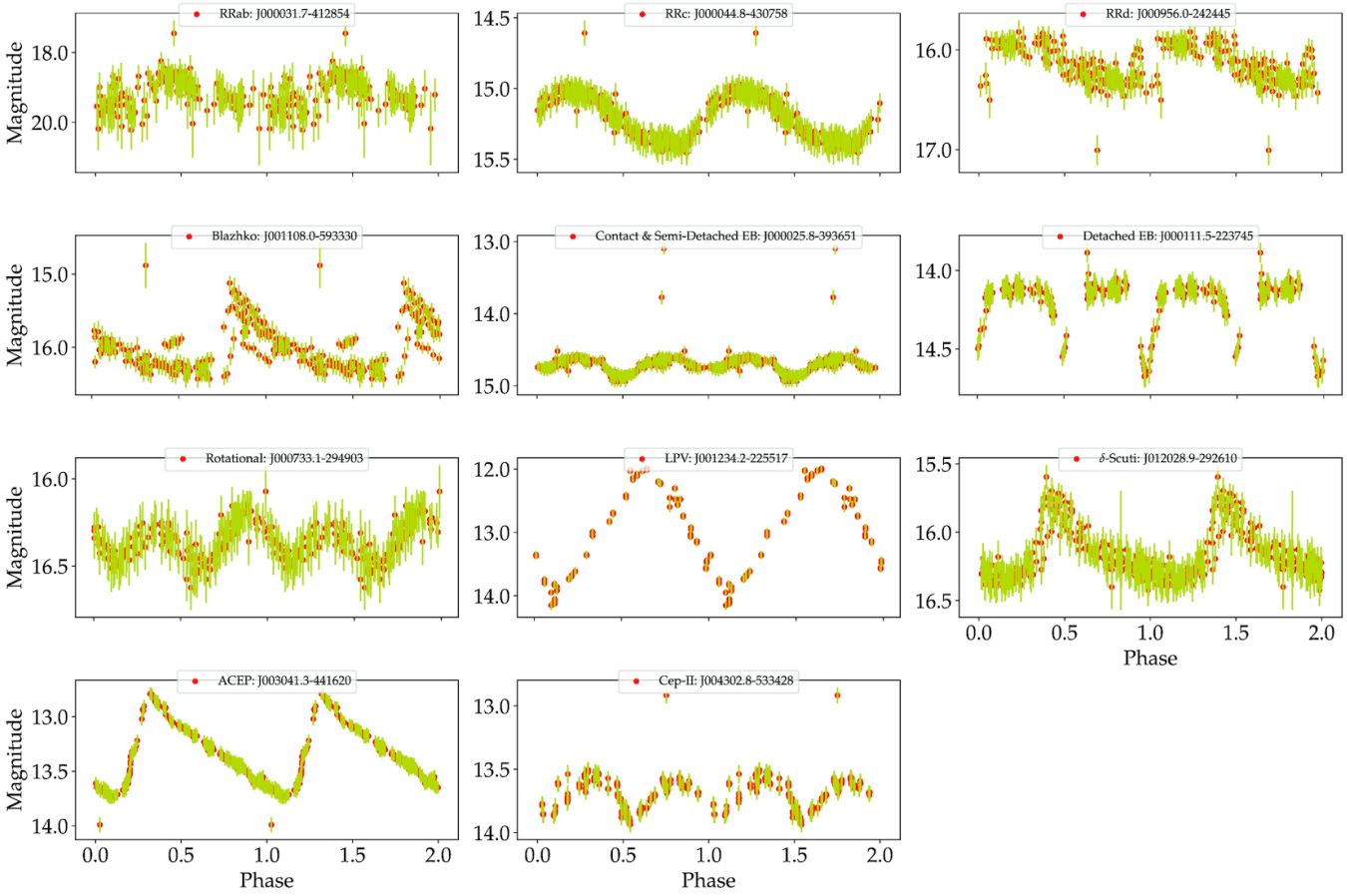
[DDC+17] A. J. Drake, S. G. Djorgovski, M. Catelan, M. J. Graham, A. A. Mahabal, S. Larson, E. Christensen, G. Torrealba, E. Beshore, R. H. McNaught, G. Garradd, V. Belokurov, and S. E. Koposov. The catalina surveys southern periodic variable star catalogue. *Monthly Notices of the Royal Astronomical Society*, 469(3):3688–3712, 05 2017.

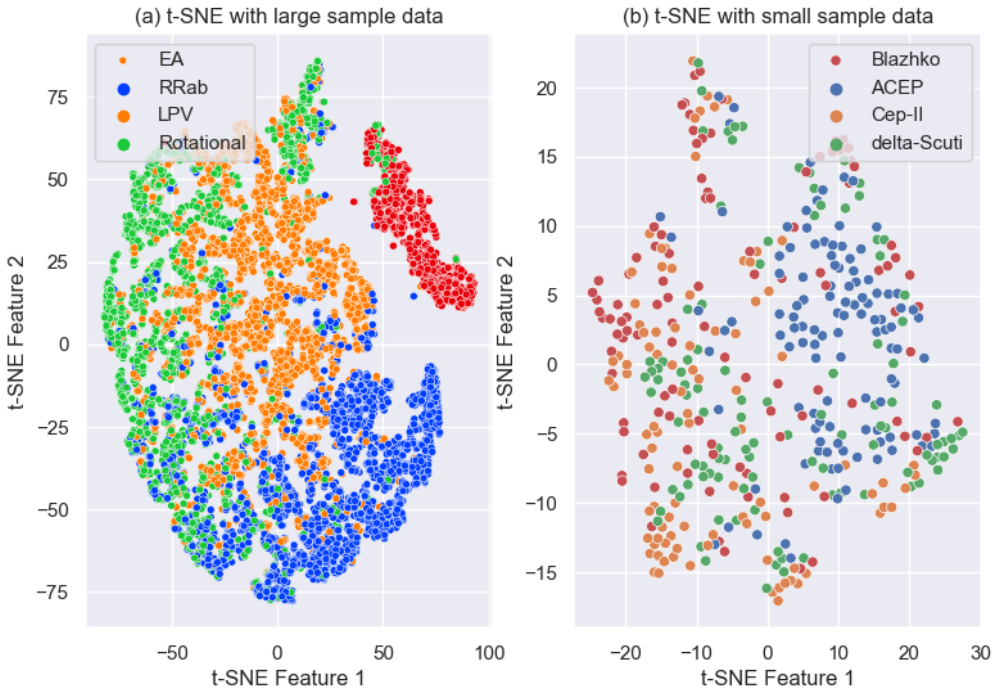
[HLSM19] Zafiirah Hosenie, Robert J Lyon, Benjamin W Stappers, and Arrykrishna Mootoovaloo. Comparing multiclass, binary, and hierarchical machine learning classification schemes for variable stars. *Monthly Notices of the Royal Astronomical Society*, 488(4):4858–4872, 07 2019.

[KBJ16] Dae-Won Kim and Coryn A. L. Bailer-Jones. A package for the automated classification of periodic variable stars. *Astronomy amp; Astrophysics*, 587, Feb 2016.

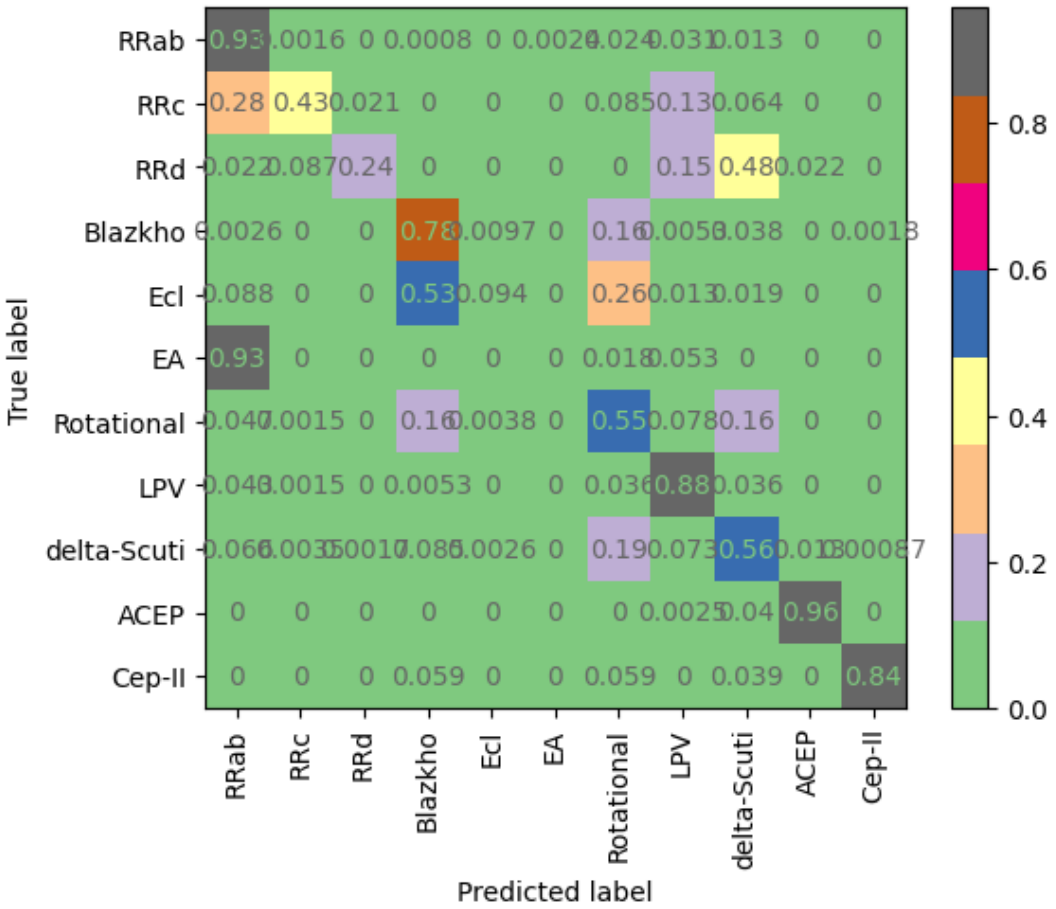
[PCPP22] R Pantoja, M Catelan, K Pichara, and P Protopapas. Semi-supervised classification and clustering analysis for variable stars. *Monthly Notices of the Royal Astronomical Society*, 517(3):3660–3681, 09 2022.

(a) Examples of folded light curves from the CRTS for the various types of variable stars considered

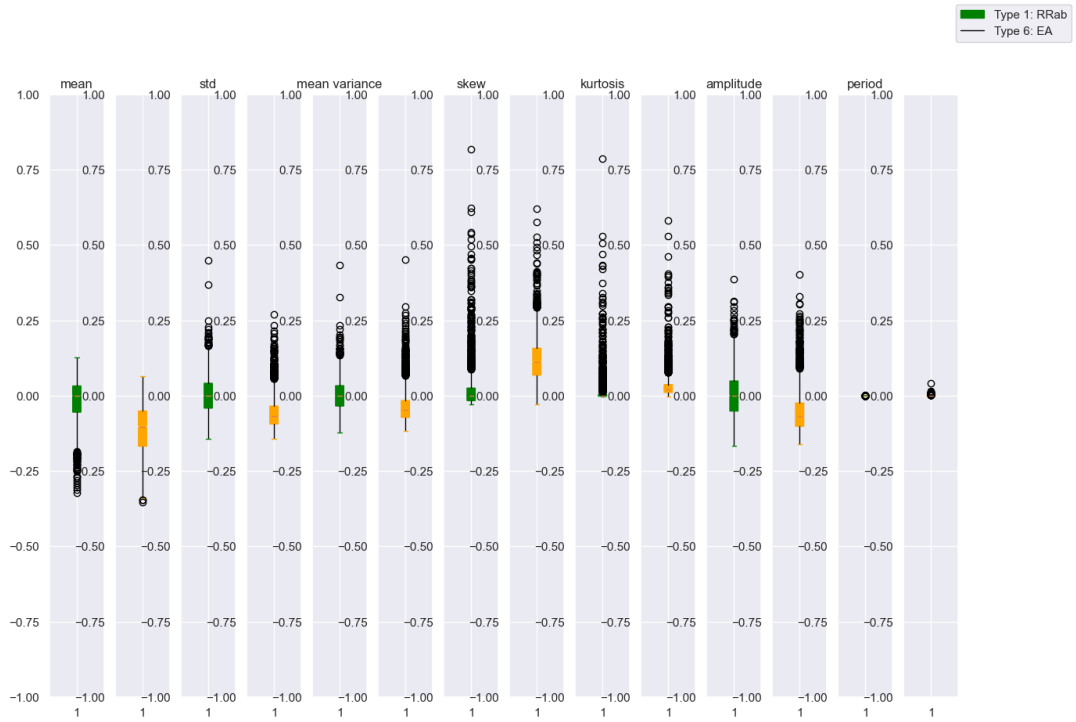




(b) (i) Shows the t-Distributed Stochastic Neighbour Embedding (t-SNE) visualization after normalization with large sample classes (RRab, EA, Rotational, and LPV) red dots are LPV. The classes are decently well separated in the embedded space. (ii) Displays t-SNE visualization for the small sample size classes. There is no distinct separation in the data.



(a) A normalized confusion matrix for the RF Classifier for multi-class classification using optimal hyper-parameters from a stratified and randomized grid search cross-validation. Trained on 70% of the data and evaluated on the other 30%. Poor performance on classes with under-represented labels.



(b) Box-and-Whisker Plots for two types of stars (RRab, EAs) against all features.