Classification of Variable Stars
Machine Learning, Report 2
Dr Raeisi/ Dr Rahvar

Mahdi Abdollahi, Ariana Haghju, Nooshin Torabi

May 2020

1 A summary of what we did

After submitting our first report, we started searching for feature extraction packages related to our data. We had to find a package that supports data with one band magnitude (I). We found a package called "feets"[1] and used it for feature extraction.
We had to use a small sample of our data (20,000 stars), because it would take a lot of time to extract all of 600,000 stars we had in our data.
We used a portion of "ECL", "Cep" and "RRLYR" from LMC stars and all the stars in two classes "ACep" and "T2Cep" from all targets. Because the latest 2 types are really small groups and our classifiers would be biased.
Then we selected 5 classifiers and in the following we will explain each algorithm's performance.
In all the process we tried to maximize both precision and recall. And so our main "score" was "f1_weighted". Because we are dealing with a multi-class classification with different weights for each class.

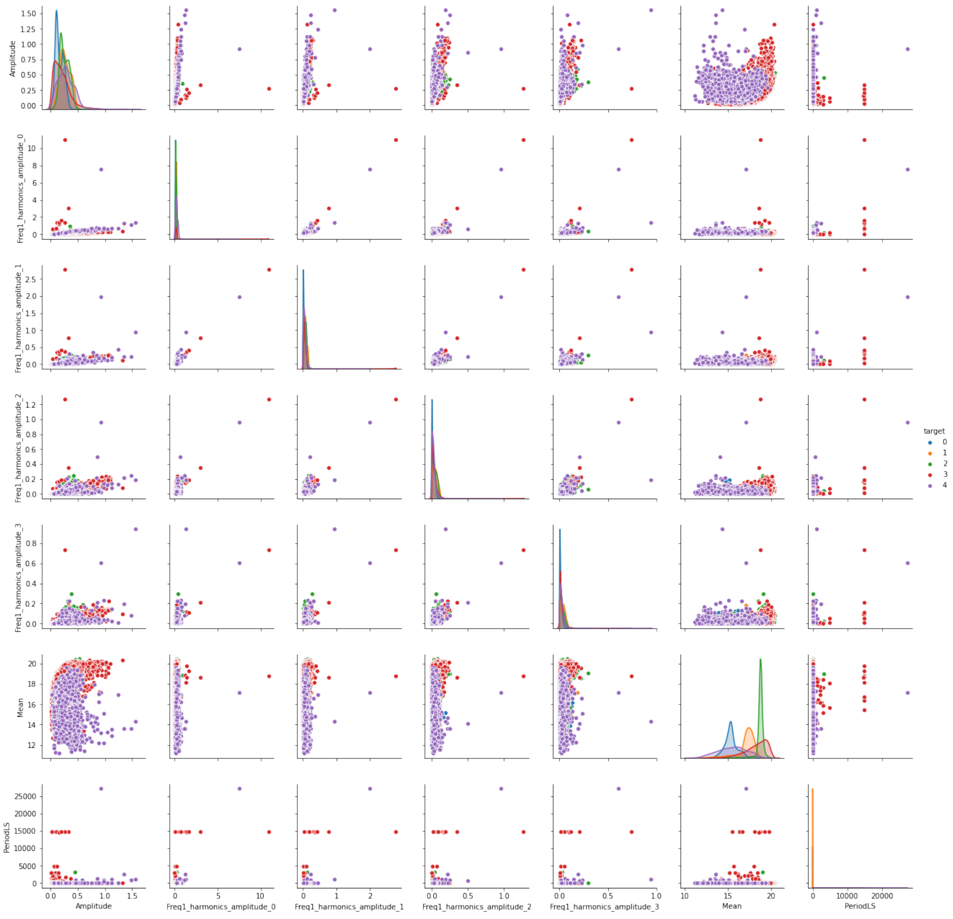


Figure 1: Pair Plot

2 Feature Extraction

Using feets package and one of the papers[2] we mentioned in our first report, we extracted 7 features for each star using their time series. These seven features are shown in table(1)
We used pair plot to see the distribution of variables and the relation between them. (figure(1))

Feature	Description
Amplitude	Half of the difference between the median of the maximum 5% and the median of the minimum 5% magnitudes
Mean	The mean magnitude of I band
PeriodLS	Period of star using LombScargle method
Freq1_harmonics_amplitude_0	Amplitude of the first harmonic of the first frequency component
Freq1_harmonics_amplitude_1	Amplitude of the 2nd harmonic of the first frequency component
Freq1_harmonics_amplitude_2	Amplitude of the 3rd harmonic of the first frequency component
Freq1_harmonics_amplitude_3	Amplitude of the 4th harmonic of the first frequency component

Table 1: Features we used for classification

3 Classification

3.1 Metric

We used f1_weighted for our scoring method, first because in our problem, both recall and precision are important and we decided that a weighted average of both, considering the unbalanced weights of our classes, is a good choice.

3.2 Random Forest

The first algorithm we used, was Random Forest. By using default hyper-parameters, we didn't get a good result. mostly because the "class_weight" was None by default and our data is not evenly distributed between the five classes. So we used gridsearch to optimize other hyper-parameters. For some hyper-parameters we also used validation curve to be sure of the exact optimized number.
The best result we could get, is shown in table (2). Finally we plotted the Learning Curve and also feature importance (figure(2)) plot using this classifier.

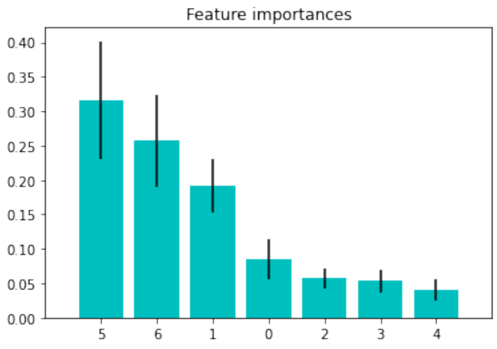


Figure 2: Feautures in order of importance: Mean, PeriodLS, Freq1_harmonics_amplitude_0, Amplitude, Freq1_harmonics_amplitude_1, Freq1_harmonics_amplitude_2, Freq1_harmonics_amplitude_3

Target	Precision	Recall	f1 score
ACep	.75	.7	.72
ECL	.95	.94	.94
RRLYR	.96	.96	.96
Cep	.9	.94	.92
T2Cep	.87	.88	.88

Table 2: Classification Report using Random Forest Classifier, weighted average of f1score = 0.94

By use of feature importance plot in RF, we chose 3 more important features to train the classifier. But we didn't get a good result and didn't work more on that.
Looking at Learning curve, we find that we are not facing an over fitting problem, but more data might have enhanced the performance of our model.
By Random Forest we reached a score of 0.94. Our problem was with the two small classes. As far as we know other attempts to classify variable stars consisting Type 2 Cepheids and anomalous Cepheids by means of traditional methods has resulted in a score like ours[2].
Learning Curve and Confusion Matrix are shown in section "Results" along other classifiers.

3.3 Ridge

Our results using Ridge classifier after tuning the model are shown in table (3).
Even after fine tuning, we couldn't enhance the score more than 0.81. By using OneVsRest classifier, the score reached 0.82.
Looking at Learning curve(figure(4)) for Ridge classifier, we find that the sample we used was enough and we're good on bias and variance.

Target	Precision	Recall	f1 score
ACep	.20	.42	.27
ECL	.92	.74	.82
RRLYR	.87	.88	.88
Cep	.54	.92	.68
T2Cep	.85	.55	.67

Table 3: Classification Report using Ridge Classifier, weighted average of f1score = 0.81

3.4 SVM

Our results using SVM classifier after tuning the model are shown in table (4).
Again by looking at learinig curve (figure(4)), we find that the size of sample is fine for SVM classifier and both the training and validation score have reached the same point. So no overfitting!
We also tried OneVsRest classifier with SVM too. The weighted f1score enhanced by 0.01. Table(4) is the OneVsRest results with SVM as estimator.

Target	Precision	Recall	f1 score
ACep	.53	.83	.65
ECL	.95	.91	.93
RRLYR	.94	.95	.95
Cep	.90	.95	.92
T2Cep	.89	.86	.87

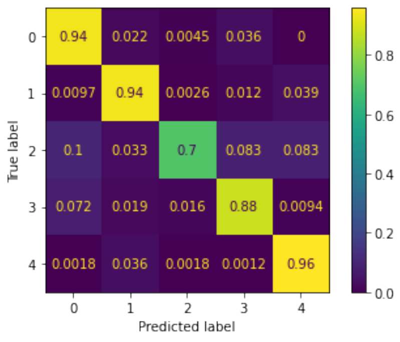
Table 4: Classification Report using SVM Classifier, weighted average of f1score = 0.93

3.5 QDA

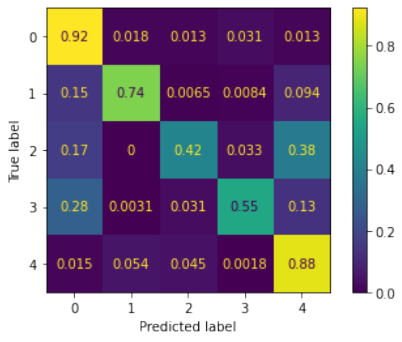
Our results using QDA classifier before(!) tuning the model are shown in table (5).
Our worst result was given by QDA classifier. Fine tuning the model, made the score even worse so here we just included the results with default hyper-parameters.

Target	Precision	Recall	f1 score
ACep	.15	.32	.21
ECL	.97	.59	.74
RRLYR	.74	.96	.84
Cep	.64	.91	.75
T2Cep	.80	.46	.59

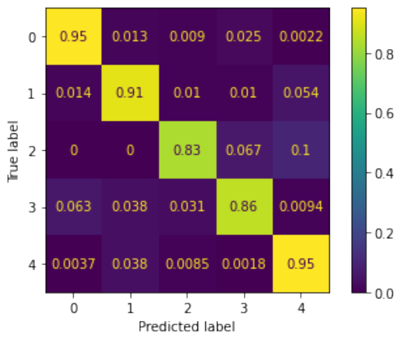
Table 5: Classification Report using QDA Classifier, weighted average of f1score = 0.76



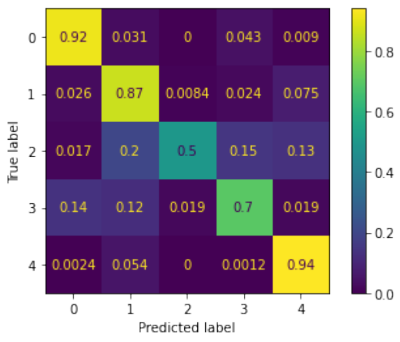
(a) Random Forest Classifier



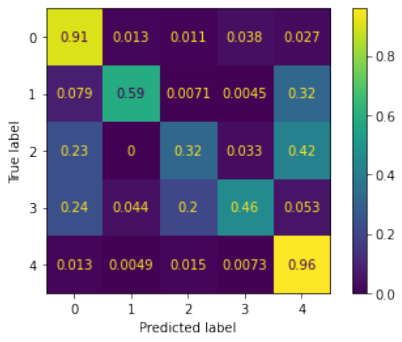
(b) Ridge Classifier



(c) SVM Classifier



(d) KNN Classifier



(e) QDA Classifier

Figure 3: Confusion Matrix

3.6 KNN

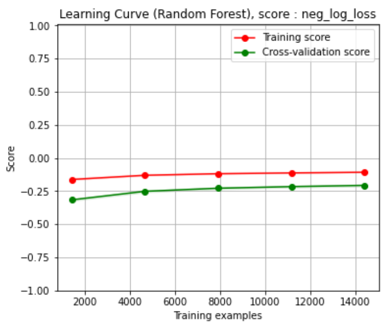
Our results using KNN classifier after tuning the model are shown in table (6).
Like Random Forest, more data could have helped us. After Fine Tuning we reached a total weighted f1 score of 0.88, which makes us KNN our third best algorithm!

Target	Precision	Recall	f1 score
ACep	.61	.50	.55
ECL	.90	.87	.88
RRLYR	.92	.94	.93
Cep	.82	.92	.87
T2Cep	.77	.70	.73

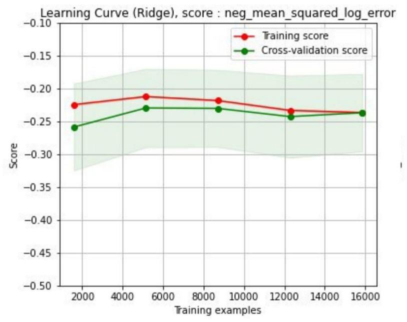
Table 6: Classification Report using KNN Classifier, weighted average of f1score = 0.88

4 Results

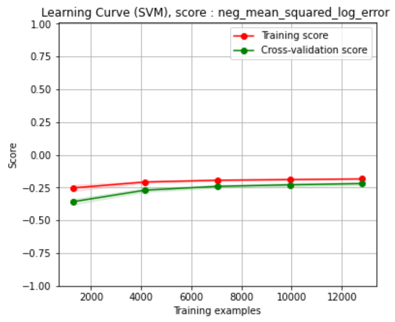
Our best estimator was Random Forest with score 0.94. If we had more data it might have been even better, analyzing the learning curve.
SVM (using OneVsRest) was the second best classifier with score 0.93. Using more data might have caused overfitting problem with this classifier so we can't expect more from this algorithm.
KNN (third best algorithm) could have worked better with more data.
Ridge and QDA were far from good results and the related learning curves show that more data couldn't help much.
All the results are shown in table(7)



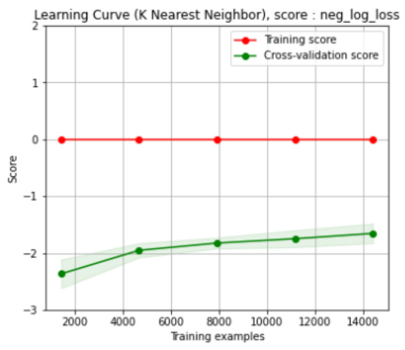
(a) Random Forest Classifier



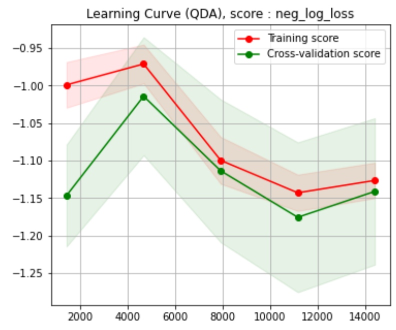
(b) Ridge Classifier



(c) SVM Classifier



(d) KNN Classifier



(e) QDA Classifier

Figure 4: Learning Curves

Classifier	Metric(f1_weighted)	Training time	Prediction time
RF	.94	27.6s	290ms
SVM	.93	18.9s	953ms
KNN	.88	20ms	30.5 ms
Ridge	.81	20.6ms	600μs
QDA	.76	18.5ms	2.34ms

Table 7: Table of final results

5 What every member of the group did:

First Mahdi worked on feature extraction by the package we had found. When the dataframe was prepared, he worked on SVM classifier.
Ariana worked on KNN and also QDA classifier.
I (which is Nooshin) worked on Random Forest and Ridge Classifier and wrote the report.
and again Here is the result!

References

[1] JB Cabral et al. "From FATS to feets: Further improvements to an astronomical feature extraction tool based on machine learning". In: *Astronomy and Computing* (2018).

[2] Dae-Won Kim and Coryn Bailer-Jones. "A Package for the Automated Classification of Periodic Variable Stars". In: *Astronomy & Astrophysics* 587 (Dec. 2015). DOI: 10.1051/0004-6361/201527188.