IDENTIFY VARIABLE STARS FROM KEPLER DATA USING MACHINE LEARNING

BY

JAYASINGHE ARACHCHIGE NIRANDI SADEEPA SANDUNI JAYASINGHE

A dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science Special Degree in Computational Physics

of the

University of Colombo, Sri Lanka

DECEMBER 2018

DECLARATION

I certify that this dissertation does not incorporate without acknowledgement, any material previously submitted for a Degree or Diploma in any university and to the best of my knowledge and belief it does not contain any material previously published or written or oral communicated by other person except where due reference is made in the text.

....

J. A. N. S. S. Jayasinghe

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my advisors Prof. Chandana Jayarathne, Department of Physics Univreisty of Colombo and Mr. Janaka Adassuriya research scientist at Arthur C. Clerk Institute for Modern technologies (ACCIMT) for the continuous support of my study and related research, for his patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisors and mentors for my research project.

Besides my advisor, I would like to thank Mr. Dharshana Kasthurirathna for their insightful comments which incented me to widen my research from various perspectives and to all my lectures at Department of Physics and UCSC for introducing me into machine learning and data science world and its applications in natural sciences.

Finally I would like to thank my family and friends who always encourage me with their support in my life.

ABSTRACT

A development of machine learning techniques to automate complex manual programs is a time relevant research in astrophysics. The growth of the availability of the data demands the need of faster, accurate automating methods of the extracting information and analyzing them. Machine learning algorithms play an impressive role in modern technology and used to address these computation and automation problems in many fields. Purpose of this project is to implement a machine learning method to address the problem automating of identifying variable stars to cope with the increasing data availability. This automated classification is built upon 6 types of star classes Beta Cephei, Delta Scuti, Gamma Doradus, Red Giants, RR Lyrae and RV Tarui using Radom Forest Classification algorithm with 19 features extracted from light curves. Machine learning models which achieve accuracy beyond 80% were considered to be successful models and our implementation achieved an accuracy of 86.5%. This study could be extended into identifying more star classes by training it on extended dataset.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

UPSILoN	Automated Classification for Periodic Variable Stars using Machine Learning
AI	Artificial Intelligence
CNN	Convolutional Neural Network
BC	Beta Cephei
DS	Delta Scuti
GD	Gamma Doradus
RG	Red Giant
RR	RR Lyrae
RV	RV Tauri

1. INTRODUCTION

Starts with varying brightness or luminosity with the time are identified as variable stars and they are considered to be very important stellar objects in astrophysics. They are used in determining the both external and internal properties such as distances, age, internal structure and future evolution of the far galaxies as well as their behavior over the time. Therefore it is obvious that researching variable stars creates the path to understanding the universe.

The prime challenge that facades in these studies is identification of variable stars from astronomical archives of big data objects as well as distinguishing them to different types of variable stars in an effective method. Ordinarily this has been doing manual programing with fairly less accuracies and highly dependable on the instrumentation as well as the data type.

With the advancements in machine learning ideas and algorithms this process has taken significant turn and become a vast research topic. In the recent present, machine learning and data analytics techniques have been applied into many areas of science, especially in astronomy. These techniques have been a very successful in handling large volume of astronomical data in simulations and astronomical calculations. Clustering and classification is one of the main applications in machine learning and many commercial and scientific applications have been developed with acceptable high level accuracies and precisions.

K- Nearest neighbor method and k-d trees are two of the successful classification methods used in astronomy forecasts such as identifying candidates for quasars at high redshift. (Kremer, Stensbo-Smidt, Gieseke, Pedersen, & Igel, 2017) Artificial Neural Network based on nonlinear and multidimensional regression is another procedure which has been applied in estimating stellar atmospheric parameters from SDSS/SEGUE spectra. (Re Fiorentin, et al., 2007) In this project, decision Tree machine learning algorithm is using in classification of selected variable star types. It has been used in many other classification systems applied in various field like finance, medical, robotics as well as astronomy and extremely popular for its accurate results.

Main purpose of this project is to address the problem of extracting useful features from corresponding light curve data and determine its type of variability depending on the extracted features within acceptable prediction accuracy($\geq$ 80%). Basically 6 different types of intrinsic variable stars from Kepler mission have been selected to apply this method of identification which uses 19 features that are extracted from their light curve data available and test the accuracies of the outcome.

1.1 Variable Stars

Stars with changing brightness over time are identified as variable stars. This brightness variation can be initiated due to its internal or external activities such as nuclear reaction taking place inside the star or due to another stellar companion.

They are classified into two groups depending on the nature of the brightness variation as, intrinsic and extrinsic. Intrinsic variable stars change their brightness by physical changes in the star or stellar system while extrinsic starts variation caused by eclipse of its stellar companion or its rotation. All these variables can be classified further into classes depending on many different features as follows. This work is based light curve gathered from Kepler mission which, belongs to following six types of intrinsic variable star groups. The model is designed and trained to classify the following six types of stars upon the features mined from time series light curve data.

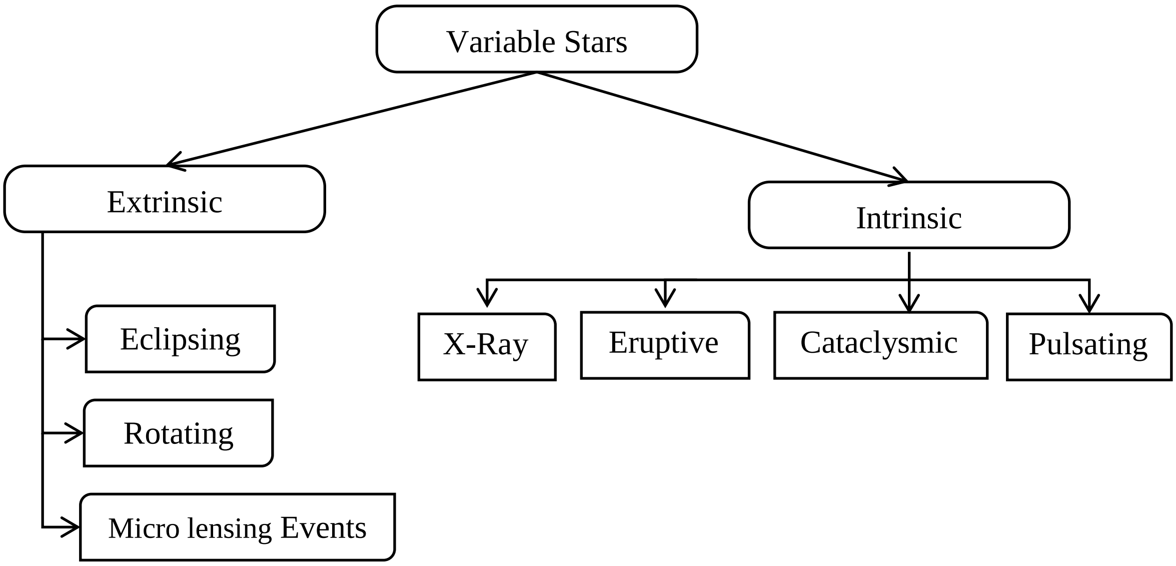

Figure 1.1 Variable star classification hierarchy based on physical characteristics

1.1.1 Beta Cephei

Naturally they are hot blue-white stars belongs to the spectral class B, also known as Beta Canis Majori. Apparently some confuses these type stars as Cepehids variables which they are not. Rapid luminosity variation arises by its pulsation in surfaces due to iron abundance at high temperature of 200,000 K that results high pressure thrusts the layers of the surface in and out.

1.1.2 Delat Scuti

Consists of several sub classes namely SX Phoenicis, oscillating Ap stars and PMS (pre-main sequence) delta scuity stars. Occasionally they are even called dwarf Cepheid too. Both non-radial and radial pulsations in cooperates to the variability. They are considered to be very important variables which are used in determination of distances to Globular clusters, galactic centers etc.

1.1.3 Gamma Doradus

Young stars with non- radial pulsations in surface which responsible for the variation of luminosity was discovered very recent past as another kind of variables.

1.1.4 Red Giant

As it describes in the name itself they are considered to be giant stars with intermediate mass and visible in red – orange - yellow visibility region. Their cores are made of hydrogen that undergoes nuclear fusion reaction to form helium just like our own sun. The few solar granules in the photospheres are accountable in the luminosity variation of these giants.

1.1.5 RR Lyrae

Often found in abundance in globular clusters. They are used in distance measurements and to study the evolution and birth of the globular clusters. The variations are caused by the temperature and as well as nuclear reactions which take place in the surfaces and the core of the star.

1.1.6 RV Tauri

Radial pulsations make changes in luminosity in these yellow- red giants. Presence of the double wave frequencies can be identified in these types of stars. They are divided into two sub classes depending on light curve properties as a and b types of RV Tarui stars.

1.2 Machine Learning

The general definition of machine learning is, “an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed” (What is Machine Learning? A definition). These techniques focus on analyzing input data and recognizing significant patterns and information to learn and access important feature for future use. This learning is done by automatically all by the computer itself.

There are two basic learning methods in machine learning, supervised learning and unsupervised learning. Supervised learning techniques are applicable when the learning data set is labeled. In this method of learning, computer is programmed to learn by known dataset and trained to apply the patterns recognized upon the dataset in future works at the end of learning phase. When the dataset is neither labeled nor classified system is train using unsupervised learning techniques. This type is mostly used when drawing inferences from dataset to describe and get a good understanding.

Machine learning covers several areas in statistics and AI. They are classification, regression, clustering, ranking, dimension reduction and density estimation. Classification is done to distinguish or label an object into one of the predefined types. Regression comes from statistical inferences and it is used in forecasting real value for a given instance (object). If there is a need of partitioning given big dataset to find the no of partitions and draw conclusions about data repository, clustering methods are used. Finding order under given conditions for ordinal dataset can be done by ranking algorithms. There can be many parameters related to an object and measuring all these quantities and modeling can be very difficult while the no of parameters increases. Dimension reduction can be applied in these situations to reduce the amount of time, cost and effort from wasting into unnecessary parameters. Density estimation of a sampled dataset can be done by machine learning algorithms as well.

These machine learning algorithms enables us to handle large quantities of data and generate meaningful information with less effort, time with high accurate results. It enhances the productivity of computations and guarantees a good result depending on the training dataset. Supervised learning algorithms are used in this work to train the model which is used to distinguish different stars using its light curve features.

2. OVERVIEW OF LITERATURE

2.1 Past and present researches conducted

Classification and analysis of stellar objects has been a major challenge in astronomy over the past few decades since the volume of the available data is increasing obligations to many successful missions conducted by pioneers in space observations. Machine Learning and statistical analysis techniques such as Kohonen self-organizing maps, Bayesian mixture-model classifier, SVMs and Gaussian mixture models etc. have been using in past decade in addressing this problem.

UPSILoN (Dae-Won & Coryn, 2015) is one of the several software packages which have been developed within the last few years, which is capable of classifying stars into different star classes including variable and non-variable using light curve data as the input irrespective of its survey specific characteristics. OGLE and EROS-2 variable star data repositories were used in training this classifier while it was tested on MACHO, ASAS and Hipparcos datasets.

(Richards, et al., 2011) It’s another machine learning framework that is used to automate the process of extracting data from light curves and identifying the type of variability. Statistical and mathematical methods of extracting important features from light curve data are defined and a thorough analysis of the results in general as well as for specific classes. Random forest classifier has been used in classification of star types as to give an in depth explanation about the features extracted by computing correlations.

2.2 Feature List

Both periodic and non-periodic features were used to do the Classifications. Light curve data are sequence of brightness observations, therefore non- periodic features can be derived from statistical time series analysis. Parameters which describe the light curve in a reliable and meaningfully are elaborated in both (Lopes & Cross, 2016) (Feature Analysis for Time Series) literature articles.

Periodic features were determined based on the generalized Lomb-Scargle method (Zechmeister & Kürster, 2009) which was built upon Fourier transformation and least square method. This analytical solution is applicable to time series data with uneven sample intervals. Main frequency or the period of the light curve is estimated by the best fit function for the defined range of frequencies based on the time series data as follows for a time series with N no of observations of y_i at time t_i with error σ_i.

Fitting a full sinusoidal function given by,

$$y(t) = a\ cos\omega t + b\ sin\omega t + c$$

For a given period ω by minimizing the residuals of the data points and the fitted curve using least square fitting.

$$\chi^2 = \sum_{i=1}^{N} \frac{[y_i - y(t_i)]^2}{{\sigma_i}^2} = W \sum w_i [y_i - y(t_i)]^2$$

$$\text{Where } \ w_i = \frac{1}{W\,\sigma_i^2} \qquad \left(W = \Sigma \frac{1}{\sigma_i^2} \qquad \Sigma w_i = 1\right)$$

Power spectrum will be obtained as follows for sampled frequencies obtained by DFT.

$$p\ (\omega) = \frac{\chi_o^2 - \chi^2(\omega)}{\chi_o^2}$$

Other periodic features were estimated for the relevant time series by the outputs of the period estimations and by results.

2.3 Random Forest Classifier

Random forest classification is a powerful and resilient framework which is based on decision trees. It is an efficient and scalable classifier that captures complicated patterns in the feature set without over-fitting into the training dataset and outliers. It fits several no of decision trees on partitions of the training dataset and takes the average to optimize the accuracy. The final prediction is determined by the majority votes which picks the most favorable output generated by constructed the trees. Definition of the random forest classifier is given as follows in (Breiman, 2001).

"A random forest is a classifier consists of a collection of tree-structured classifiers $\{h\ (\ x, \Theta_k), k = 1, \ldots\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x"

Trees in the random forest are built upon bootstrap samples of equal size to the original dataset. As the sampling is done with replacements same instances can be repeated in the sample while some are left out. These mislaid data instance called Out-of –Bag (OOB) are used to evaluate the prediction error. Classification proceeds by partitioning the drawn sample recursively into more trees or nodes based on randomly selected attribute from the previously chosen attribute list. This will repeat until a terminal node reach which contains a single type of objects. Each of these trees which are constructed provides a prediction for the type and final prediction is determined to be the type with the highest aggregation.

This machine learning classifier can even be applied to find the order of importance for attribute or feature list. It can be used to reduce the feature list which will be helpful to reduce the computational cost for large complex data repositories.

3. THEORY

3.1 Derivation of Generalized Lomb-Scargle Periodogram and periodic Feature extraction

For sinusoid with a constant model,

$$y(t) = a\, cos\omega t + b\, sin\omega t + c \qquad (1)$$

Chi square fit for the data and (1) to obtain the squared differences of the fit and data point

$$\chi^2 = W \sum w_i [y_i - y(t_i)]^2 \qquad (2)$$

In order to minimise the error function χ^2 must be minimized. Minimum of χ^2 can be determine by taking the partial derivatives with respective to the coefficients of (1).

$$0 = \partial_a \chi^2 = 2W \sum w_i [y_i - y(t_i)]\, cos\omega\, t_i \qquad (3)\text{-}1$$

$$0 = \partial_b \chi^2 = 2W \sum w_i [y_i - y(t_i)]\, sin\omega\, t_i \qquad (3)\text{-}2$$

$$0 = \partial_c \chi^2 = 2W \sum w_i [y_i - y(t_i)] \qquad (3)\text{-}3$$

Above set of equation (3) will give conditions for the minimum in linear equations.

$$a = \frac{YC\,.\ SS - YS\,.CS}{D} \qquad (4)\text{-}1$$

$$b = \frac{YS.CC - YC\,.CS}{D} \qquad (4)\text{-}2$$

Where

$$Y = \sum w_i y_i \qquad (5)$$

$$C = \sum w_i \cos\omega\, t_i \qquad (6)$$

$$D(\omega) = CC.SS - CS^2 \qquad (7)$$

$$\text{YY} = \widehat{YY} - Y\,.Y \qquad \widehat{YY} = \sum w_i {y_i}^2 \qquad (8)$$

$$YC(\omega) = \widehat{YC} - Y.C \qquad \widehat{YC} = \sum w_i y_i \ cos\omega t_i \tag{9}$$

$$YS(\omega) = \widehat{YS} - Y.S \qquad \widehat{YS} = \sum w_i y_i \sin \omega t_i \tag{10}$$

$$CC(\omega) = \widehat{CC} - C.C \qquad \widehat{CC} = \sum w_i \ cos^2\omega t_i \tag{11}$$

$$SS(\omega) = \widehat{SS} - S.S \qquad \widehat{SS} = \sum w_i \ sin^2\omega t_i \tag{12}$$

$$CS(\omega) = \widehat{CS} - C.S \qquad \widehat{CS} = \sum w_i \ cos\omega t_i \ sin\omega t_i \tag{13}$$

Amplitude of the best fitting function at frequency ω is given by$\sqrt{a^2 + b^2}$. Considering all above minimum χ^2 can be written as follows.

$$\frac{\chi^2}{W} = \sum w_i[y_i - y(t_i)] \ y_i - \sum w_i\left[y_i - y(t_i)\right] y(t_i)$$

$$\sum w_i[y_i - y(t_i)] \ y(t_i) = a \sum w_i[y_i - y(t_i)] \ cos\omega \, t_i + b \sum w_i[y_i - y(t_i)] \ sin\omega \, t_i + c \sum w_i[y_i - y(t_i)] = 0$$

Therefore,

$$\frac{\chi^2}{W} = \widehat{YY} - a \, \widehat{YC} - b \, \widehat{YS} - cY$$

$$\frac{\chi^2}{W} = \widehat{YY} - Y.Y - a(\widehat{YC} - Y.C) - b(\widehat{YS} - Y.S)$$

$$\frac{\chi^2}{W} = YY - aYC - bYS$$

Substituting from equation set (4)

$$\frac{\chi^2}{W} = YY - \frac{SS \, .YC^2}{D} - \frac{CC. \ YS^2}{D} + 2\frac{CS \, .YC \, .YS}{D}$$

By normalizing the χ^2

$$p\,(\omega) = \frac{1}{YY \, .D} \ [SS \, .YC^2 + \ CC. \ YS^2 - 2CS \, .YC \, .YS] \tag{14}$$

3.2 Statistical Non-Periodic Feature List

1) Mean: Average Magnitude of the light curve.

$$\bar{M} = \frac{\sum m_i}{N} \tag{15}$$

2) STD: Standard Deviation

$$\sigma = \frac{1}{N-1}\sum_i (y_i - \bar{y})^2 \tag{16}$$

3) Median: middle value of the sorted Magnitudes.

4) Skew

$$Skewness = \frac{N}{(N-1)(N-2)}\sum_{i=1}^{N}\left(\frac{m_i - \bar{m}}{\sigma}\right)^3 \tag{17}$$

5) Kurtosis: measurement of the peak of the magnitude distribution.

6) Mean Variance

$$Mean\,Variance = \frac{STD}{Mean} \tag{18}$$

7) Stetson K: Robust kurtosis measure taken using Magnitudes and errors.

$$\delta_i = \sqrt{\frac{n}{n-1}}\frac{m_i - \bar{m}}{m_{err}} \tag{19}$$

$$Stetson\ K = \frac{\left(\frac{\sum |\delta_i|}{N}\right)}{\sqrt{\frac{\sum \delta_i^2}{N}}} \tag{20}$$

8) Max Slope: Maximum absolute slope between two consecutive observations.

9) Amplitude: Half the difference between the median of the maximum 5% and minimum 5% magnitudes.

10) Median absolute deviation

$$median\ absolute\ deviation = median\ (\,|Magnitude - median(Magnitude)\,|) \tag{21}$$

11) m $_{p\,10}$: 10% percentile of slope of the light curve.

12) m $_{p\,90}$: 90% percentile of slope of the light curve.

13) Variability Index: This parameter is used to identify trends in data points or their independence. Following is the variability index for uneven sampled data.

$$\eta_e = \bar{w}\,(t_{N-1} - t_1)^2 \, \frac{\sum_1^{N-1} w_i (m_{i+1} - m_i)^2}{\sigma^2 \sum_1^N w_i} \tag{22}$$

$$w_i = \frac{1}{(t_{i+1} - t_i)^2} \tag{23}$$

3.3 Classifier performance

There are metrics to measure the performance of a classifier based on the testing output of the trained classifier.

Table 3.1 Classifier Outcomes for each class

Predicted Class	**True Class**	
	Positive	*Negative*
Positive	**True positive (TP)** (Instance is positive and it is classified as a true positive)	**False positive (FN)** (Instance is negative and classified as true)
Negative	**False negative (FN)** (Instance is positive and classified as negative)	**True negative (TN)** (Instance is negative and predicted as negative)

1) Recall/ Sensitivity

$$Recall = \frac{TP}{TP + FN} \tag{24}$$

2) Precision

$$Precision = \frac{TP}{TP + FP} \tag{25}$$

3) Accuracy

$$Accuracy = \frac{TP + TN}{Total\ inputs} \tag{26}$$

Recall and precision are calculated for each class in this case for all 6 classes which are considered. Accuracy is calculated generally for the all classed depending on the outcomes for testing data.

4) F_1 Score

$$F_1 = 2\ \times \frac{precision\ \times recall}{precision + \ recall} \tag{27}$$

This gives a value in the range of 0 – 1 and calculated for each class separately. Use to measure the performance of the classifier.

4. METHODOLOGY

4.1 Understanding the Application Domain

First stage of any scientific procedure is to get a profound understanding of the problem. This includes understanding the application domain, objectives of the task, end results as well as the past and current procedures done to solve the problem.

Variable stars are very interesting topic in astrophysics yet over the years physicists have given less attention in changing the manual processes or methods of analysing and extracting data by applying various techniques introduced by computer science, only in very recent present physicist have considered these techniques in scientific world and its developing in a rapid manner every day. Intersection of machine learning techniques and variable stars were identified as the main domains in this project and the literature review was done in both fields to get a good understanding on the background.

Objectives of the project were built and altered based on the literature review, were established. Relevant research papers were referred to find the most suitable methodology to success the project. Depending on the specified objectives computation needs were researched. Python 3.0 was selected out of many popular candidate machine learning language frameworks and tools such as C/C++, JavaScript, R and MATLAB. Its machine learning modules are simple, elegant, consistent and also it handles large computations simply by few lines in few seconds. On top it's an open source language. Python 3.0 language was studied to do the coding related to the project. (Intro to Python for Data Science) (Introduction to Python for Data Science) (VanderPlas, 2016) Few specific existing python modules were studied in depth to get grasp knowledge in using them under project specific environment.

CNN and Random Forest machine learning classification techniques were selected to do a feasibility study. From this it was discovered, CNN method is biased to the instrumentation specification and also it is hard to obtain images with the same quality and specifications. Therefore it was decided that, random forest classification using time series data is more applicable in this matter considering the availability of data and ability to extract data which are not biased by instrumentation specifications.

4.2 Creating the Target Dataset

Identifying the suitable type of data is important in this phase. In this case two types of input data was considered, image files of light curve data and time series data. First attempt was to do classification based on the light curve image data using CNN but this was ruled out due to instrumentation specification effects and unavailability of quality dataset.

Dataset was extracted from Kelpler missions in .dat and .pow text file format. Full dataset consists of 583 light curve data files of 6 different types of stars as mentioned in Introduction. These files were recorded in many different metric systems and different parameters were recorded. 540 Out of 583 of data files were selected to carry out the project, which had all the necessary parameters (Time, Corrected Flux, and Corrected Flux error) recorded in acceptable metric systems.

Table 4.1 Dataset Composition

Star Class	No of Light Curves
BC	120
DS	64
GD	127
RG	113
RR	90
RV	26
Total	***540***

4.3 Pre-processing

This phase can be divided into three processors reduction, cleaning, integrating. This phase was the most time consuming phase after literature review. Selected data files were in different formats, in different metric systems as well as with outliers, missing data fields and were in string data type.

Different procedures in python pandas module was tested on these raw data files to find a way to extract only necessary data fields and bring them into one metric systems without changing the data by such as automatic truncations. All these manual programs failed to extract data in a proper manner without merging two different fields or truncating time and flux in this raw file types, .dat and .pow. Therefore all these files were converted into CSV file type using spread sheet software MS excel. In this conversion process, data fields were separated manually by dropping out the unnecessary headers and meta data, renaming the fields, converting them into numeric data type without truncations in values and finally by filling in missing data fields with 'NaN'.

After converting all the files into usable dataset they were fed into the python code written to carry out reduction process which was the first lines in the feature extraction code "feature.py" given in the appendix A. Data fields were checked for unbalanced data columns, no of data points and the data points with a magnitude that were beyond $\pm 3\sigma$ level were dropped to clean out the outliers. Finally the time scale was fixed to start from 0 time value except for different astronomical date systems such as baycentric Julian dates.

4.4 Extracting Features and Transforming Data

Pre-processed data is used to extract meaningful parameters on the light curve data which is later fed to the classifier to train the model. This feature extraction is automated and the python code is given in the appendix.

For the training and testing process first the features were extracted separately and saved into a file. This feature extraction was divided into two levels periodic and non-periodic feature extraction. Periodic features are based on the period estimation of the time series dataset and its procedure. This period estimation is done according to the Lomb-Scargle period estimation procedure. There were three in built-modules, SciPy (scipy.signal.lombscargle), astropy (Lomb-Scargle Periodograms), astroML (Example of Lomb-Scargle Algorithm) to estimate this Lomb-Scargle peridogram. The UPSILoN package also had an open source code published specifically to estimate short period of variable stars. UPSILoN period estimate code was adjusted according to this project specification. Periods were estimated but it was failed since mostly it detected noise as the true period which later on affected the classifier training process.

In order to improve the poor period estimation and reduce the risk period was estimated in two ways and both periods were used in training and testing processes.

I. Astropy LombScargle in- built module (Period_1),

II. Lomb Scargle function coded from the scratch based on (Zechmeister & Kürster, 2009) (Period_2)

Both way of estimation was better than the UPSILoN functions which were used previously. (see Appendix A for the code feature.py)

4.5 Training the Classifier

Random Forest classifier in sklearn python module was used to define the classifier and were trained by 70% of dataset and tested upon rest of the 30%. Random Forest parameters were selected as 700 and 10 respectively for no of trees (t), no of features randomly selected at each node of trees (m). The rest of the specifications of the Random forest classifier are given below.

```
RandomForestClassifier (bootstrap=True,
             class_weight=None, criterion='gni',
            max_depth=None, max_features=10, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=700, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

The training had to be done two times due to the poor accuracy achieved in the first time caused by the period estimation. Second attempt was successful and achieved an accuracy > 80% for all the classifiers which were trained. Also feature importances were calculated as a part of the training step to identify the feature ranks or the contribution toward the classification from each feature. All the scripts and functions used in this stage are given in Appendix A under "classifier_Training.py" python script.

4.6 Testing the Trained Classifier

Trained classifier was tested on the testing portion of the dataset which consist of 163 rows of data corresponding to same no of light curves. Test outcomes were classified using confusion matrix and color maps. The classifier performance metrics were calculated to evaluate the trained classifier using sklearn available modules and syntaxes. Analysis was done on the testing outcomes and finally the trained classifier is dumped and used in predicting the type of the new star by extracting features from their light curves. (classifier_Training.py)

5. RESULTS AND ANALYSIS

5.1 Extracted Feature Analysis

Following observation were illustrated from the full dataset before spliting into test and train datasets.

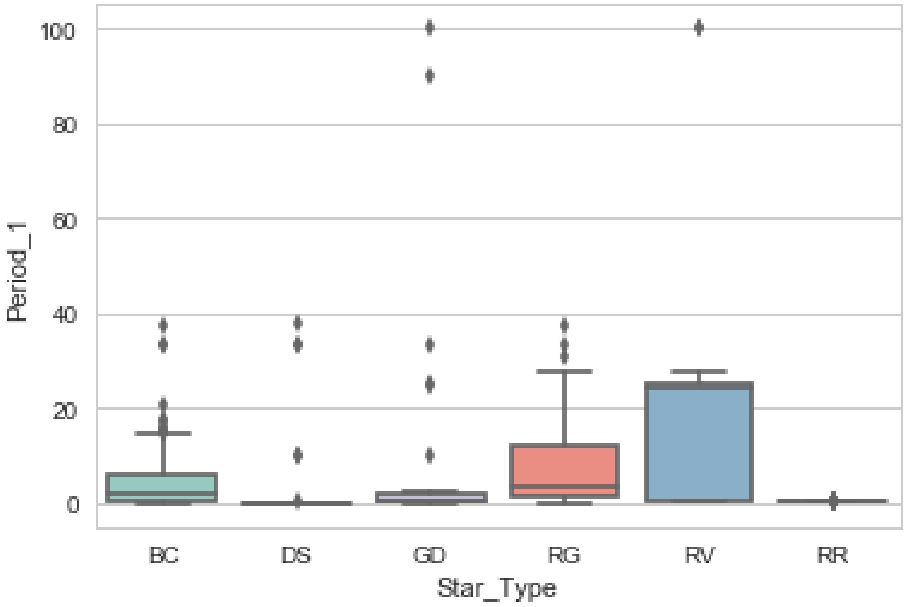

Figure 5.1 Period 1 distrbution

Table 5.1 Period 1 Statistics

Statistic	BC	DS	GD	RG	RR	RV
Mean	4.956827	1.993686	3.132539	7.084616	0.544112	20.611567
STD	7.053866	7.496141	12.388318	8.361377	0.066972	26.176088
Minimum	0.121676	0.048103	0.224319	0.190249	0.470602	0.511802
25%	0.359813	0.054570	0.376545	1.407662	0.488103	0.513187
50%	1.947484	0.067597	0.664061	3.281253	0.514274	24.147055
75%	6.097800	0.142671	1.801545	12.250866	0.561803	25.083090
Maximum	37.304178	37.83961	100.00000	37.160096	0.682197	100.00000

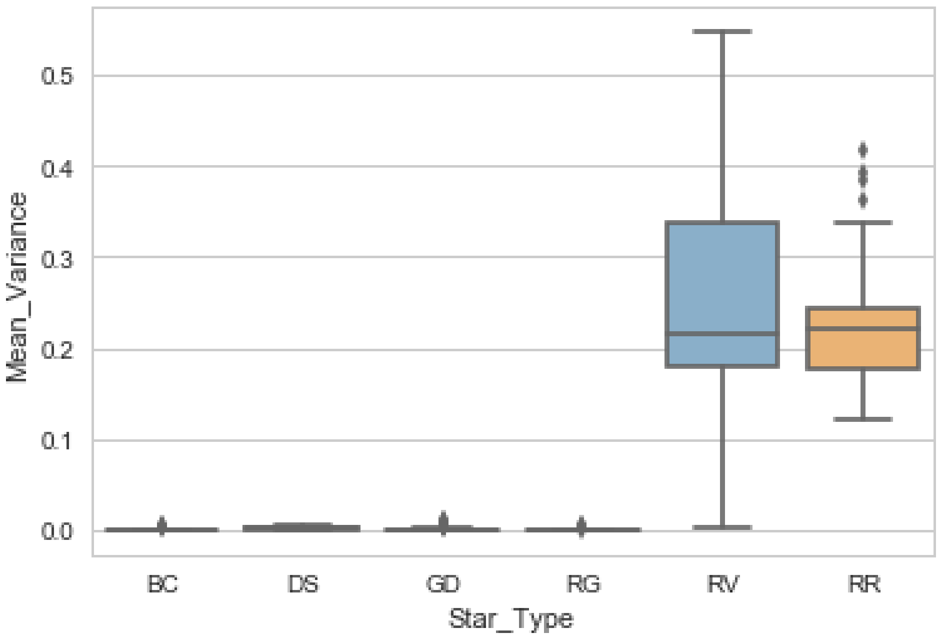

Figure 5.2 Mean Variance distribution

Table 5.2 Mean Variance Statistics

Statistic	BC	DS	GD	RG	RR	RV
Mean	0.000712	0.002158	0.001206	0.000848	0.222609	0.255924
STD	0.000518	0.001726	0.001659	0.000453	0.062890	0.126664
Minimum	0.000383	0.000214	0.000149	0.000405	0.121603	0.001751
25%	0.000509	0.000508	0.000437	0.000712	0.176296	0.178661
50%	0.000647	0.001636	0.000660	0.000819	0.219484	0.216437
75%	0.000743	0.003172	0.001072	0.000901	0.243593	0.337939
Maximum	0.004431	0.005860	0.010612	0.004043	0.416183	0.546282

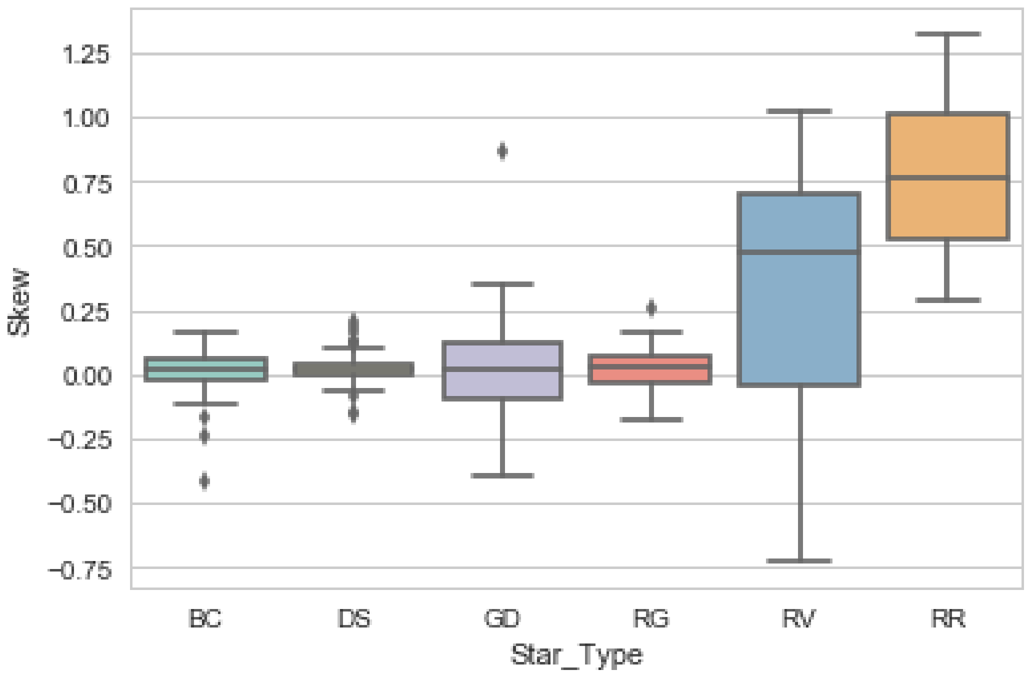

Figure 5.3 Distribution of Skew

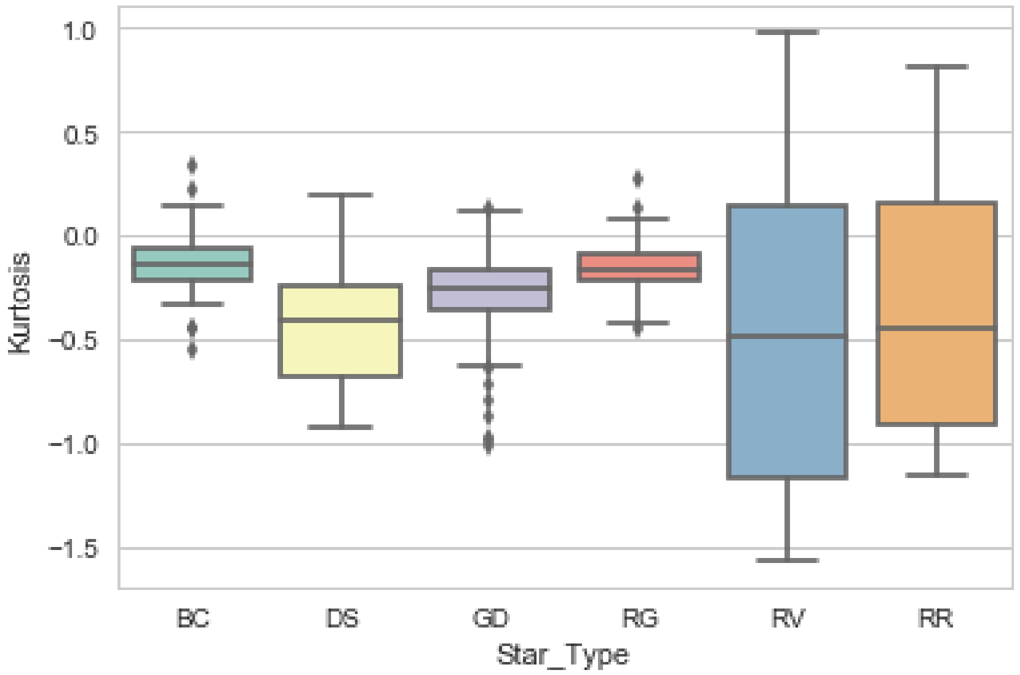

Figure 5.4 Distribution of Kurtosis

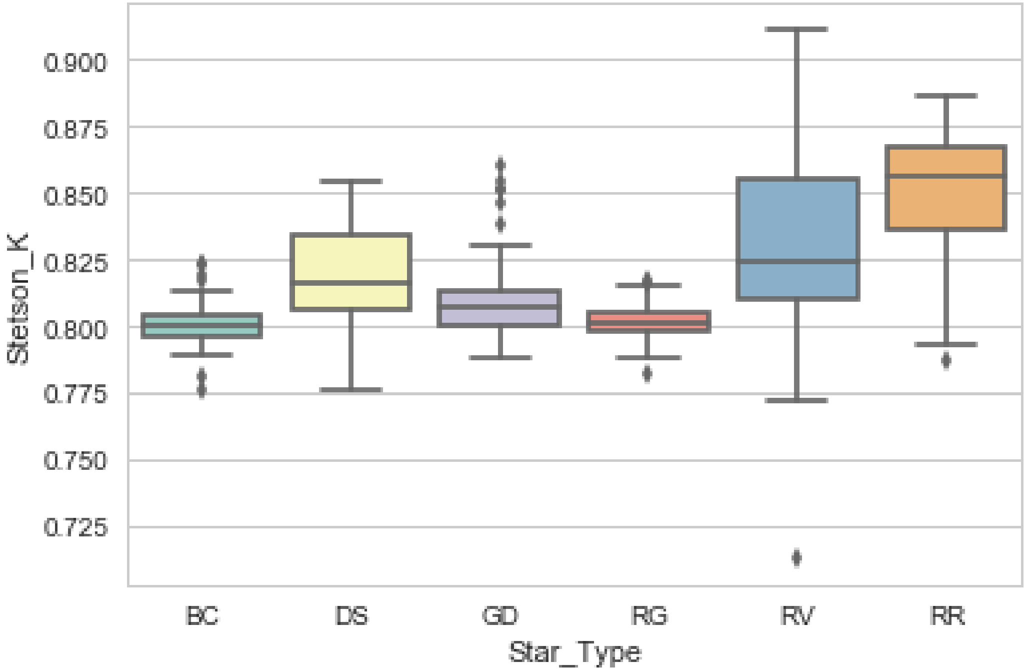

Figure 5.5 Distribution of Stetson K

5.2 Classifier Results and Analysis

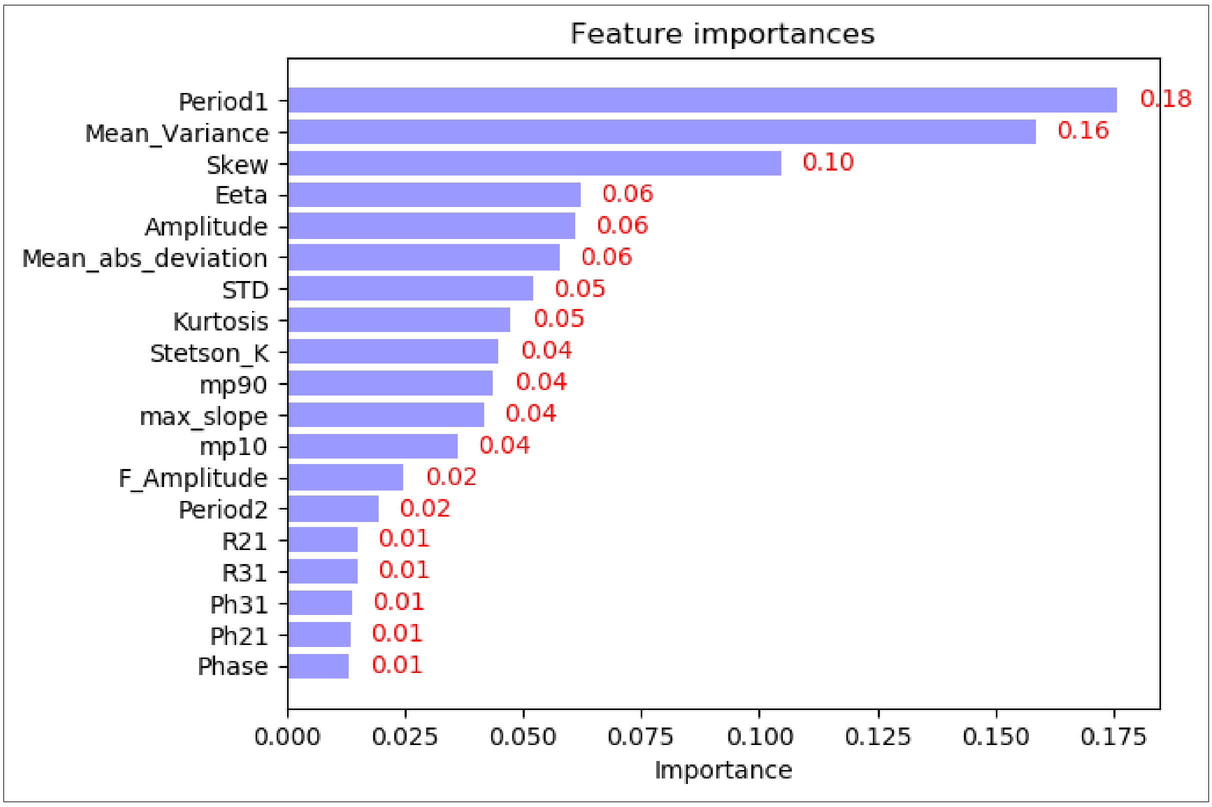

Figure 5.6 Ranked List of Features by decreasing importance measured using Random Forest

Feature importance was evaluated from the training process based on the training dataset. This helps to identify the contribution from each feature to the classification process.

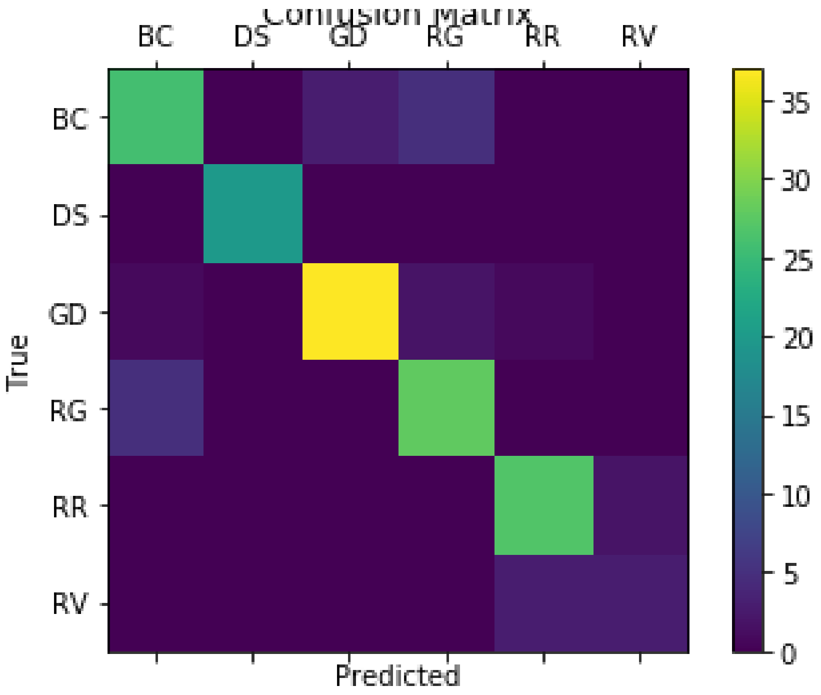

Figure 5.7 Confusion Matrix for obtained for testing dataset

Table 5.3 Confusion Matrix in Tabular Form

Predicted Class	**BC**	**DS**	**GD**	**RG**	**RR**	**RV**
True Class						
BC	26	0	3	5	0	0
DS	0	20	0	0	0	0
GD	1	0	37	2	1	0
RG	5	0	0	28	0	0
RR	0	0	0	0	27	2
RV	0	0	0	0	3	3

Confusion matrix describes the no of misclassifications in cross tabular or heat map corresponding to the cross table of predicted and true class.

Table 5.4 Classifier Performances

	Precision	Recall	F1-Score	support
BC	0.81	0.76	0.79	34
DS	1.00	1.00	1.00	20
GD	0.93	0.90	0.91	41
RG	0.80	0.85	0.82	33
RR	0.87	0.93	0.90	29
RV	0.60	0.50	0.55	6
Avg / Total	0.86	0.87	0.86	163
Accuracy % = 86.5				

6. DISCUSSION

Machine Learning processes are emerging techniques in many fields in contemporary days and astrophysics is one of those fields. Many processes that were claimed to be impossible to automate in past, were made possible by these techniques from reducing the complexity and avoiding the need to code every step of instructions. It includes identifications, classification, estimations, forecasting to extracting information on objects. Therefore using machine learning to identify and classify variable stars is pertinent approach in many ways.

Section 5.1 elaborate the results obtained based on light curve extracted feature values. Period is considered to be the most important of all features considered throughout this project and it is determined using two ways based on Lomb-Scargle period estimation. Method that uses in-built python module was more successful than the function which was coded from the scratch in estimating period. This difference in estimation is caused from the selection of period range and the step size to calculate the equation (14). But still there were instances in all types, where estimated value for period was not acceptable. Figure 5.1 visualizes the distribution of periods estimated from the astropy LombScargle function for the full dataset. It shows a clear variation of range of the period depending on the type of the star as well as over estimations. Table 5.1 gives a brief statistical description of the estimated period values versus star type. It shows a clear picture of the period distributions by quartile values. 50% of the values are spread below 1.947484 days and 75% of values lie below 6.097800 days. Likewise for DS, GD, RG, RR, RV 75% of periods are distributed below 0.142671, 1.801545, 12.250866, 0.561803, and 25.083090 respectively.

Second most important feature of the list is verified as the mean Variance which is calculated by equation (18). Figure 5.2 and Table 5.2 describes the parameter values for each type of the star both visually and tabular form. Clear discrepancy can be observed in the boxplot of mean variance, between the types RV, RR and BC, DS, GD, RG types. For further clarifications among the types table 5.2 can be used. It clearly shows the ranges of the mean variance distribution based on the class. Ranges are [0.000383, 0.004431], [0.000214, 0.005860], [0.000149, 0.010612], [0.000405, 0.004043], [0.121603, 0.416183], [0.001751, 0.546282] corresponding to BD, DS, GD, RG, RR, RV types. RG type has the lowest distribution range while RV has the widest spread. According to mean value of this parameter, star types can be ordered in increasing mean variances as, GD, DS, BC, RG, RR and RV.

Rest of the boxplots represented in figures 5.3 to 5.4 envisions the distributions of skew, kurtosis and Stetson K parameters. Just like in previous parameters there is a clear difference in ranges of the distribution of RV and RR type among other types. They have to have larger ranges or wider spread compared to other 4 types of the star classes.

Results obtained from the training and testing process is included in the latter section of chapter 5. This includes the feature importance and classifier performances evaluation. 540 light curves belongs to 6 types of stars are used in this project and the out of that 377 light curves were used in training process while the rest 163 were used in testing process.

Figure 5.6 displays the feature importance bar chart with a numerical value for importance of the corresponding feature. According to this evaluation which is based on the Random Forest classifier, the most significant feature is Period with a significance value of .18 followed by other parameters in decreasing significance Mean Variance 0.16, Skew 0.10, Eeta 0.06, Amplitude 0.06 etc. this implies that period is the main factor when determining the type of the star. Also the equal importance value represents that those features have equal contribution in determining the star type. This do not necessarily indicates that they are correlated features. The main advantage of evaluating this feature importance is to reduce the size of the dataset by eliminating the least significant features. There are methods of doing this elimination process. This elimination is important when expanding this project into more classes in order to reduce the computation complexity.

Testing phase is carried out to evaluate the performances of the trained classifier. This evaluation is done based on the confusion matrix and the performance metrics corresponds to the classifier. Confusion matrix is represented in figure 5.7 and table 5.3. This illustrates the no of correctly predicted classes and misclassifications. Diagonal elements of the grid gives the no of correctly classified instances while off diagonals gives the misclassifications. 8 instances of BC stars were misclassified as 3 GD and 5 RG, 4 instances of GD was misclassified as 1 BC, 2 RG, 1 RR, 5 instances of RG has misclassified as BC, 2 instances of RR were misclassified as RV and 3 instances of RV were misclassified into RR. These observations convey that there's a tendency to misclassify RR and RV into each other's classes or there are many similarities in RR and RV star classes. Classifier metrics are calculated based on this confusion matrix following equations (24) to (27).

Precision, Recall F1-Score, for each class as well as overall average values and accuracy achieved by the classifier is represented in table 5.4. DS star class has achieved the 100% of highest precision, recall and F1 score among all the classes for this classifier which means that all the instances belongs to this class were correctly classified. Lowest performance metrics were obtained by the RV class and they are 0.60, 0.50, and 0.55 respectively. All the other classes have achieved precision≥ 80%, recall≥ 76%, F1-Score≥ 79%. In generally all average performance metrics are above 85%. Overall Accuracy of the classifier is 86.5 %.

Estimating the period of the star corresponding to the light curve data was the major issue which was faced in undertaking this project. Since the expected range of the periods of the time series data of many classes were different most the in-built python modules were inappropriate for this project because there were no way to define a general range for all the types of classes. Still there were unexpected results for the period estimations and the classifier was trained including those results which directly impacts the accuracy of it. Period estimation should be enhanced further to extend the limits of the classifier to identify more star classes. Also some instances had long gaps or data were missing for a considerable amount of time gap which may have caused shifts in the feature estimation.

Also there were some issues with gathering data and pre-processing stage since the available data were in different metric systems, various file types, missing data which reduced the dataset. Generally more data means more accuracy therefore using larger dataset advantages sometimes.

In order to improve this machine learning method and expand this to next level, period estimation method has to be improved and as high quality dataset with more instances is needed.

7. CONCLUSIONS

- Machine learning techniques provide easy and effective approaches in astrophysics and astronomy to cope with the increasing analyzing demands and the increase of data availability.
- These techniques are fast and demands less time compared to manual process the quality of the classifier strongly determined by the quality of the training dataset.
- Random forest is a powerful algorithm, allows doing classifications of star types based on its light curve features with remarkable performances.
- Feature importance estimation available in Random Forest classifier is useful in expanding the system and the most significance features in classification can be identified.
- Drawbacks of period estimation reduce the accuracy leading to misclassify similar types of stars.
- In presence of a quality dataset it is possible to train a model with an accuracy $\geq 80\%$ using random forest machine learning algorithm.

REFERENCES

Breiman, L. (2001). Random Forest. *Kluwer Academic Publishers*, 5-32.

Dae-Won, K., & Coryn, A. L.-J. (2015). A package for the Automated Classification of periodic Variable Stars. *Astronomy & Astrophysics*.

Example of Lomb-Scargle Algorithm. (n.d.). Retrieved 08 2018, from astroML.org: http://www.astroml.org/book_figures/chapter10/fig_LS_example.html

Feature Analysis for Time Series. (n.d.). Retrieved 10 10, 2018, from Institute for Applid Computational Science Harvard School os Engineering and Applied Sciences: http://isadoranun.github.io/tsfeat/FeaturesDocumentation.html

Intro to Python for Data Science. (n.d.). Retrieved 07 23, 2018, from Data Camp: https://www.datacamp.com/courses/intro-to-python-for-data-science

Introduction to Python for Data Science. (n.d.). Retrieved 08 2018, from edx.org: https://courses.edx.org/courses/course-v1:Microsoft+DAT208x+2T2018/course/

Kremer, J., Stensbo-Smidt, K., Gieseke, F., Pedersen, F. S., & Igel, C. (2017). Machine Learning and image analysis for astronomy. *IEEE Intelligent Systems*, 16-22.

Lomb-Scargle Periodograms. (n.d.). Retrieved 08 2018, from astropy.org: http://docs.astropy.org/en/stable/stats/lombscargle.html

Lopes, C. F., & Cross, N. (2016). New Insights into Time Series Analysis. *Astronomy and Astrophysics*.

Re Fiorentin, P., Bailer-Jones, C., Beers, T., Lee, Y., Sivarani, T., Wilhelm, R., et al. (2007). Estimate of stellar atmospheric parameters from SDSS/SEGUE. *Astronomy & Astrophysics*, 1373-1387.

Richards, J., Starr, D., Butler, N., Bloom, J., Brewer, J., Crellin-Quick, A., et al. (2011). Machine Learned Classification of Variable Stars with Sparse and Noisy Time-Series Data. *The Astrophysical Journal*.

scipy.signal.lombscargle. (n.d.). Retrieved from SciPy.org: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.lombscargle.html

Spangler, M. R., & Bredehoeft, G. (2013, 11 14). *U.S. Energy Information Administration*. Retrieved from http://www.eia.gov/todayinenergy/detail.cfm?id=13791

What is Machine Learning? A definition. (n.d.). Retrieved 10 24, 2018, from Expert System: http://www.expertsystems.com/machine-learning-definition/

Zechmeister, M., & Kürster, M. (2009). The Generalied Lomb-Scargle periodogram. *Astronomy & Astropysics*.

APPENDIX A

“Feature.py script”

Loading the dataset to a dataFrame cleaning the data set

```
import pandas as pd
import numpy as np
from numpy.fft import *
import scipy.stats as ss
from scipy.optimize import leastsq
from astropy.stats import LombScargle
import matplotlib.pyplot as plt
from sklearn.externals import joblib
import csv
import math

df= pd.read_csv(r"C:\Users\Nirandi\Documents\Code\Data\RV\RV (22).dat.csv")

df.fillna({
    'Corrected Flux':'[A Za z]',
    'Corrected Flux error':'[A Za z]'
})
df = df.dropna()

T = np.array(df["Time"])
Mag = np.array(df["Corrected Flux"])
Err = np.array(df["Corrected Flux error"])

#fixing Time array to normal time starting from 0 days
T = T-T[0]
#cleaning the outliers
M = np.mean(Mag)
S = np.std(Mag)
index=np.where((Mag > M + 3*S) | (Mag < M - 3*S))
Mag=np.delete(Mag,index)
T=np.delete(T,index)
Err=np.delete(Err,index)

#checking the lengths
if (len(T) != len(Mag)) or (len(T) != len(Err)) or (len(Mag)!= len(Err)):
    print("The length of Date, Mag, Err is not the same.")

#checking the no of data points
if(len(T) < 100):
    print("The number of data points are less than 100")
```

Extracting stats and non-periodic features

```
N = len(T)    # no of data points
Err2 = np.array(1 / (Err**2))   # inverse error squared array
```

```
W = np.array(sum(Err2))    # W sum of inverse error squared
w = np.array(Err2 / W)    # weight factor

#simple stats
Mean = np.mean(Mag)
Median = np.median(Mag)
STD = np.std(Mag)
Skew = ss.skew(Mag)
Kurtosis = ss.kurtosis(Mag)
Mean_variance = STD/Mean

#Stetson K index evaluation
delta = (Mag-Mean) / Err
stetson_k = (np.sum(np.fabs(delta))/len(Mag)) / np.sqrt(np.sum(delta**2)/len(Mag))

#Basic quantities derived from Mag
mean_abs_deviation = np.median(abs(Mag-Median))

#Calculating the Variability index Eeta
wt = np.zeros(N-1)
m = np.zeros(N-1)
for l in range(N-2) :
    wt[l] = wt[l] + 1. / (T[l+1] - T[l])**2
    m[l] = m[l] + (Mag[l+1] - Mag[l])**2
Eeta = np.mean(wt)*(T[N-2]-T[0])**2*sum(wt*m) / (STD**2*sum(wt))

#Amplitude=Half the diffence between the median of the maximum 5% and the median of the minimjm 5%
M = np.sort(Mag)
per = round(5*N/100)
mini = np.median(M[:per+1])
maxi = np.median(M[N-per:])
Amplitude = (maxi-mini)/2

#Calculating the Max Slope
slope=np.zeros(N-1)
for l in range (N-2):
    slope[l] =slope[l]+abs((Mag[l+1]-Mag[l])/(T[l+1]-T[l]))
Max_Slope=np.max(slope)

# Calcuating mp10: 10% percentile of slopes of a phase folded light curve
m_10 = np.percentile(slope,10)

# Calcuating mp90: 90% percentile of slopes of a phase folded light curve
m_90 = np.percentile(slope,90)
```

Extracting Periodic Features Period: in days Amplitude: before pre-whitening Phase: R21=ratio of first amplitude of the fourier decomposition to second R31=ratio of first amplitude of the fourier decomposition to third

```
Tspan=T[N-1]-T[0]  #Time span of the dataset
```

```
f0=round(1/Tspan,2)   #min frequency
df=0.1/Tspan   #frequency step
S=np.zeros(N-1)
for i in range(N-2) :
    S[i]=S[i]+1./(T[i+1]-T[i]) # 1/delta T
fn=round(0.5*np.mean(S),2)   #maximum frequency
```

LombScagle function

```
frequency, power = LombScargle(T, Mag, Err).autopower(minimum_frequency=f0, maximum_frequency=fn, samples_per_peak=10)
loc = np.argmax(power)
period1 =1/ frequency[loc]


def residuals(pars, x, y, order):
        return y - fourier_series(pars, x, order)

def fourier_series(pars, x, order):
        sum = pars[0]
        for i in range(order):
            sum += pars[i * 2 + 1] * np.sin(2 * np.pi * (i + 1) * x) \
                   + pars[i * 2 + 2] * np.cos(2 * np.pi * (i + 1) * x)

        return sum

def LombScarglePeriod(t, y, sigma):
    Ttot = t[-1]
    delta = 0.1 / Ttot
    min_f = 1/Ttot
    max_f = 20
    frq = np.arange( min_f, max_f, delta)
    P = np.zeros(len(frq))

    for k in range(len(P)):

        f = frq[k]
        W = np.sum (1/sigma**2)
        w = (1/W)*(1/sigma**2)

        Y = np.sum (w*y)
        C = np.sum (w*np.cos(2*np.pi*f*t))
        S = np.sum (w*np.sin(2*np.pi*f*t))

        YYhat = np.sum (w*y**2)
        YChat = np.sum (w*y*np.cos(2*np.pi*f*t))
        YShat = np.sum (w*y*np.sin(2*np.pi*f*t))
        CChat = np.sum (w*np.cos(2*np.pi*f*t)**2)
        SShat = np.sum (w*np.sin(2*np.pi*f*t)**2)
        CShat = np.sum (w*y*np.cos(2*np.pi*f*t)*np.sin(2*np.pi*f*t))
        YY = YYhat - Y*Y
        YC = YShat - Y*C
        YS = YShat - Y*S
        CC = CChat - C*C
        SS = SShat - S*S
        CS = CShat - C*S

        D = CC*SS-CS**2

        P[k] = (SS*YC**2 + CC*YS**2 - 2*CS*YC*YS)/(YY*D)

    loc = np.argmax(P)
    Period =1/ frq[loc]

    # Fit Fourier Series of order 3.
    order = 3
    # Initial guess of Fourier coefficients.
    p0 = np.ones(order * 2 + 1)
    date_period = (t % Period) / Period
    p1, success = leastsq(residuals, p0,args=(date_period, y, order))

    # Derive Fourier features for the first period.
    # Petersen, J. O., 1986, A&A
    amplitude = np.sqrt(p1[1] ** 2 + p1[2] ** 2)
    r21 = np.sqrt(p1[3] ** 2 + p1[4] ** 2) / amplitude
    r31 = np.sqrt(p1[5] ** 2 + p1[6] ** 2) / amplitude
    f_phase = np.arctan(-p1[1] / p1[2])
    phi21 = np.arctan(-p1[3] / p1[4]) - 2. * f_phase
    phi31 = np.arctan(-p1[5] / p1[6]) - 3. * f_phase


    return Period, amplitude, r21, r31, f_phase, phi21, phi31

period2, amplitude, r21, r31, f_phase, phi21, phi31 = LombScarglePeriod(T, Mag, Err)

# Saving the Features to a CSV file

#Star=

with open(r'Features.csv', 'a', newline='') as csvfile:
    fieldnames = ['Period1','Period2','F_Amplitude','R21','R31','Phase','Ph21','Ph31','Skew',\

'Kurtosis','STD','Mean_Variance','Stetson_K','max_slope','Amplitude','Mean_abs_deviation',\
                 'mp10','mp90','Eeta']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)


writer.writerow({'Period1':period1,'Period2':period2,'F_Amplitude':amplitude,'R21':r21,'R31':
r31,'Phase':f_phase,'Ph21':phi21,'Ph31':phi31,\
'Skew':Skew,'Kurtosis':Kurtosis,'STD':STD,'Mean_Variance':Mean_variance,'Stetson_K':stetson_k,'max_slope':Max_Slope,\

'Amplitude':Amplitude,'Mean_abs_deviation':mean_abs_deviation,'mp10':m_10,'mp90':m_90,'Eeta':Eeta})
```

“ Classifier_Training.py ” Script

Feature importance is done to the extracted data from light curves to identify the importance of each feature to the classification. This is done using Random Forest Classifier.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.externals import joblib


data=pd.read_csv(r'C:\Users\Nirandi\Documents\Code\Features.csv')
#shuffel the dataset
df= shuffle(data)
r, c = df.shape

#feataure set
X = df.iloc[:,0:c-1].values

#labels
Y = df.iloc[:,c-1].values

#splitting dataset into train and test
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, )
```

Training The Classifier

```
#Fitting te Random Forest Classifier to the training set
classifier = RandomForestClassifier (n_estimators = 700, max_features = 10)
classifier.fit(X_train, Y_train)
```

Testing the Classifier

```
# predict the test set resukts
Y_pred = classifier.predict(X_test)
```

```
target_names = ['BC', 'DS', 'GD', 'RG', 'RR', 'RV']
#Confusion matrix
cm = confusion_matrix(Y_test, Y_pred, target_names)
CM = pd.crosstab(Y_test, Y_pred, rownames = ['True Class'], colnames= ['Predicted Class'])
print(CM)

#print confusion matrix
fig = plt.figure()
ax=fig.add_subplot(111)
cax=ax.matshow(cm)
plt.title('Confusion Matrix')
fig.colorbar(cax)
ax.set_xticklabels([''] + target_names)
ax.set_yticklabels([''] + target_names)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()

#feature importance
importance = classifier.feature_importances_
#print(list(zip(df.columns[0:19], importance)))

indices = np.argsort(importance)[::-1]

feature = ['Period1','Period2','F_Amplitude', 'R21', 'R31', 'Phase', 'Ph21', 'Ph31', 'Skew',
'Kurtosis', 'STD', 'Mean_Variance',\
        'Stetson_K', 'max_slope','Amplitude','Mean_abs_deviation', 'mp10', 'mp90', 'Eeta' ]
label=[]

for f in range(c-1):
#    print("\n "+ feature[indices[f]] +" %d (%f)" % (f + 1,  importance[indices[f]]))
    label.append(feature[indices[f]])

print(classification_report(Y_test, Y_pred, target_names=target_names))
Accuracy = accuracy_score(Y_test, Y_pred, normalize=True)
print("Accuracy % = ",(Accuracy*100))

plt.rcdefaults()
fig, ax = plt.subplots()

ax.barh(range(c-1), importance[indices], facecolor="#9999ff", align="center")

ax.set_yticks(range(c-1))
ax.set_yticklabels(label)

for x, y in zip(range(c-1), importance[indices]):
    plt.text(y +0.01 , x , '%.2f' % y, ha='center', va= 'center', color ='r')

ax.invert_yaxis()  # labels read top-to-bottom
ax.set_xlabel('Importance')
ax.set_title('Feature importances')

plt.show()

# Save the model
joblib.dump(classifier, 'classifier8.joblib')
```

“classifier.py”

Trained Classifier using to predict the Star Class for new dataset

```
import pandas as pd
import numpy as np
from numpy.fft import *
import scipy.stats as ss
from scipy.optimize import leastsq
from astropy.stats import LombScargle
import matplotlib.pyplot as plt
from sklearn.externals import joblib
import csv
import math

data= pd.read_csv(r"C:\Users\Nirandi\Documents\Code\Features.csv")

# Load the saved Model
model = joblib.load('classifier4.joblib')

# New inputs
X = data.iloc[:,:]

# Apply the model to predict the class
Result = model.predict(X)
print(Result)
```

APPENDIX B

Statistical Summary of Extracted Features