McMaster STATS 4CI3 Final Project

Winter 2021

Pulsar Classification

Tommy Flynn

400121496

April 27, 2021

Contents

1 Introduction [2] 1

2 The HTRU2 Dataset [1] 1

3 Classification Techniques 4

3.1 Radom Forests [5] 4

3.2 Gaussian Mixture Models [4] [6] 5

4 Results 7

4.1 Random Forest Results 7

4.2 Gaussian Mixture Model Results 8

5 Conclusions 8

References 8

1 Introduction [2]

When a massive supergiant star collapses, it forms a dense object called a neutron star. In rare cases, neutron stars can emit electromagnetic radiation from its poles that is periodically detectable on Earth due to rotation; we call this particular variant a pulsar. Pulsars are of prime interest in science and astronomy as they are the vessel on which we can study phenomena such as space-time, stellar evolution, gravitation, and the interstellar medium. In order to take advantage of these unique stars, it is paramount that we accurately separate candidates into neutrons and pulsars.

2 The HTRU2 Dataset [1]

HTRU2 is a publicly available pulsar dataset on the UCI Machine Learning Repository [3]. It is a collection of 17898 neutron stars. As mentioned above, pulsars are quite rare so there are 1639 positive examples and 16259 negative examples. Each observation has 8 features which are descriptive statistics from the integrated profile (IP) and the dispersion measure-signal to noise ratio (DM) curve. The IP of a pulsar is its unique fingerprint determined by emission pattern and rotation and the DM curve accounts for the dispersion and noise interfering with the signals as they travel to Earth. Figure 1 shows the IP and DM curve of a candidate star. IPs tend to be centered about the mean whereas DM curves tend to be rather skewed. The particular descriptive statistics used for the features are the mean, standard deviation, excess kurtosis, and skewness of both the IP and the DM curve. Respectively, these measure the central tendency, the spread, the tails compared to the standard normal distribution, and the symmetry of the curves. Lastly, we have the response variable which accounts for the type of star, neutron or pulsar.

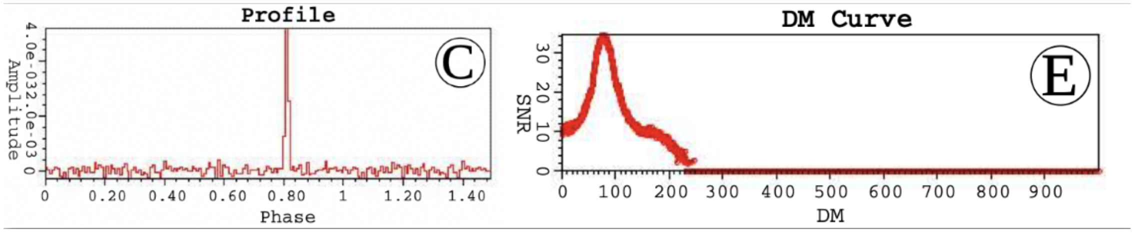
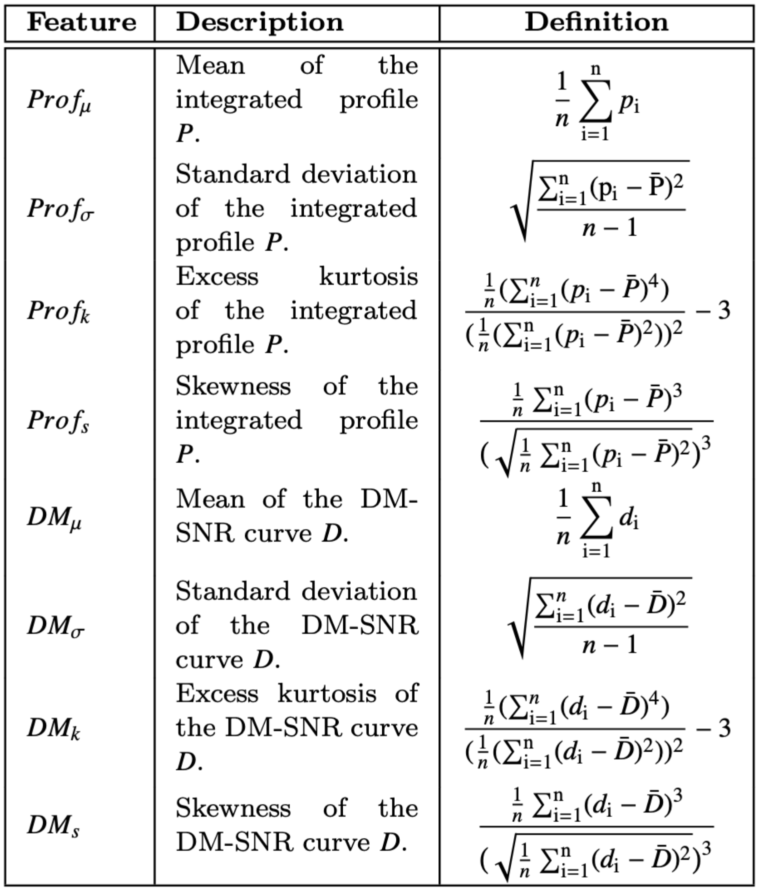


Figure 1: Candidate IP (left) and DM Curve (right) [1].

In summary, we have the following features shown in Figure 2



Feature	Description	Definition
$Prof_{\mu}$	Mean of the integrated profile P.	$\frac{1}{n}\sum_{i=1}^{n} p_i$
$Prof_{\sigma}$	Standard deviation of the integrated profile P.	$\sqrt{\frac{\sum_{i=1}^{n}(p_i-\bar{P})^2}{n-1}}$
$Prof_{k}$	Excess kurtosis of the integrated profile P.	$\frac{\frac{1}{n}(\sum_{i=1}^{n}(p_i-\bar{P})^4)}{(\frac{1}{n}(\sum_{i=1}^{n}(p_i-\bar{P})^2))^2}-3$
$Prof_{s}$	Skewness of the integrated profile P.	$\frac{\frac{1}{n}\sum_{i=1}^{n}(p_i-\bar{P})^3}{(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i-\bar{P})^2})^3}$
DM_{μ}	Mean of the DM-SNR curve D.	$\frac{1}{n}\sum_{i=1}^{n} d_i$
DM_{σ}	Standard deviation of the DM-SNR curve D.	$\sqrt{\frac{\sum_{i=1}^{n}(d_i-\bar{D})^2}{n-1}}$
DM_{k}	Excess kurtosis of the DM-SNR curve D.	$\frac{\frac{1}{n}(\sum_{i=1}^{n}(d_i-\bar{D})^4)}{(\frac{1}{n}(\sum_{i=1}^{n}(d_i-\bar{D})^2))^2}-3$
DM_{s}	Skewness of the DM-SNR curve D.	$\frac{\frac{1}{n}\sum_{i=1}^{n}(d_i-\bar{D})^3}{(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i-\bar{D})^2})^3}$

Figure 2: Feature Set. [1]

and the class of the star given by $C_i = \begin{cases} 0 & \text{if neutron star} \\ 1 & \text{if pulsar} \end{cases}$

The variables are presented in Table 1 using a 6-number summary. We can see that the features take on a variety of magnitudes and will need to be scaled. The most surprising value is the maximum DM skewness which takes on a value one order of magnitude larger than expected. In particular, we notice that the features for IP tend to be smaller than the mean IP while the features for DM tend to be larger than the mean DM. This is understandable as it matches the centered and skewed behaviour of the IP and DM curve shown in Figure 1.

Table 1: Summary of the data.

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
$Prof_\mu$	17,898	111.080	25.653	5.812	100.930	127.086	192.617
$Prof_\sigma$	17,898	46.550	6.843	24.772	42.376	51.023	98.779
$Prof_k$	17,898	0.478	1.064	−1.876	0.027	0.473	8.070
$Prof_s$	17,898	1.770	6.168	−1.792	−0.189	0.928	68.102
DM_μ	17,898	12.614	29.473	0.213	1.923	5.464	223.392
DM_σ	17,898	26.327	19.471	7.370	14.437	28.428	110.642
DM_k	17,898	8.304	4.506	−3.139	5.782	10.703	34.540
DM_s	17,898	104.858	106.515	−1.977	34.961	139.309	1,191.001

	Neutron	Pulsar	N
Class	16259	1639	17989

Next we look at the pairs plot in Figure 3 which shows that most of the features will be useful in separating each class of star despite the large number of outliers. The variables of strongest negative correlation are $Prof_\mu$ with $Prof_s$ and $Prof_k$ as well as DM_k with DM_σ. On the other hand, the variables of strongest positive correlation are $Prof_s$ with $Prof_k$, DM_s with DM_k, and DM_μ with DM_σ. In general, we see that the mean and standard deviation tend to be positively correlated with each other and negatively correlated with the kurtosis and skewness.

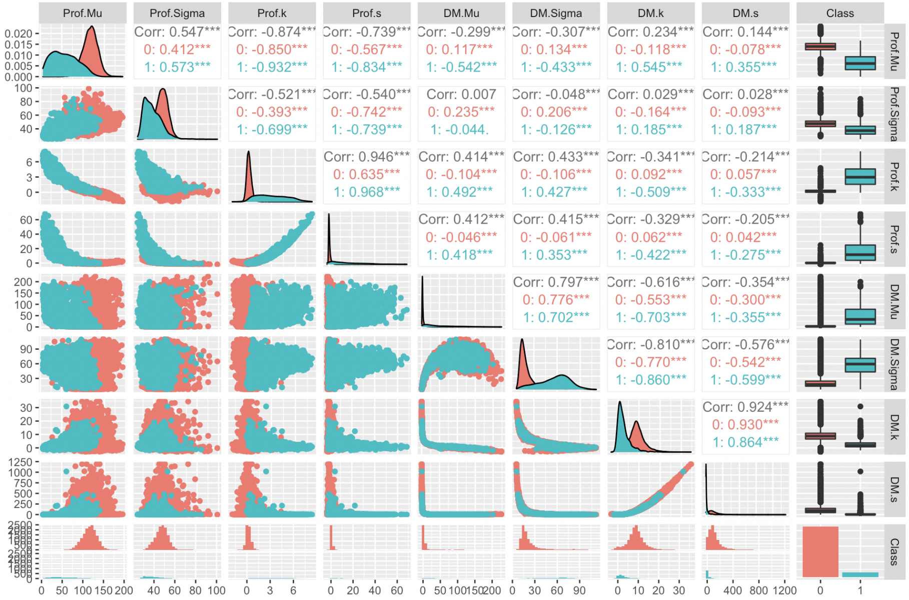


Figure 3: Pairs plot with Neutron (Red) and Pulsar (Blue).

3 Classification Techniques

In this section, we examine two methods to separate neutron and pulsar stars. The first is supervised learning algorithm known as a random forest which performs classification of the stars. The second is an unsupervised learning algorithm known as a Gaussian mixture model which performs clustering of the stars.

3.1 Radom Forests [5]

Random forests are an ensemble method meaning they are a collection of weak learners that pool together to create strong predictions. The fundamental component of this particular ensemble is the decision tree which is string of conditionals that depends on the features of the dataset. Given an observation at the root, a decision tree traverses through its list of conditional branches and provides a final class prediction at the leaves. An example of a single decision tree with two classes is shown in Figure 4.

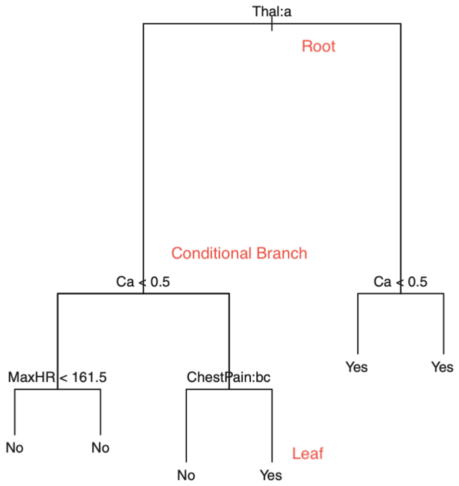


Figure 4: Decision Tree [5].

Diversity of the trees inside a random forest arises from training each individual tree on a bootstrap sample of the training data. To implement a random forest, we first split the data into two sets with 67% in training and 33% in testing. We will be using stratified random sampling which ensures the splits are representative by taking a 67% random sample from both the neutron and pulsar populations. That is, stratified random sampling accounts for the fact that only 10% of the data comes from pulsars while 90% comes from neutrons. Next, we select its hyper-parameters which are chosen to be the number of trees and the number of features. To select the optimal parameters, we apply grid search and cross-validation. Grid search traverses through the hyper-parameter space and evaluates forests at each coordinate. Our hyper-parameter space will be an 8 by 5 grid with 40 nodes representing 1 to 8 features and 100 to 500 trees. At each grid point, we will evaluate the model using 5-fold cross validation. 5-fold cross validation partitions the data into 5 equal parts where 4 are used to train the forest and the remaining fold is used for validation of the model. In this way, 5 forest's will be trained using all possible combinations of the data and the model score at this node will be the average validation error of each forest. After the grid search is complete, a final random forest will be trained on all of the training data using the optimal hyper-parameters. The final score will be given by the prediction accuracy of the best model on test data which has been left untouched during the model selection and training.

3.2 Gaussian Mixture Models [4] [6]

A Gaussian mixture model assumes that the data is generated from a finite mixture distribution where each component is a multivariate Gaussian of unknown parameters. Mathematically, the model density takes the following form

$$f(x) = \sum_{g=1}^{G} \pi_g \phi(x|\mu_g, \Sigma_g) \tag{1}$$

where π_g is the probability that an observation belongs to component g and $\phi(x|\mu_g, \Sigma_g)$ is a multivariate Gaussian of mean μ_g and covariance matrix Σ_g. To fit the model, we use the EM algorithm which provides the maximum likelihood estimates of the parameters via an iterative scheme. The EM algorithm assumes the data is of the form (x, z) where x is observed and z is latent. If $p(x, z|\theta)$ is the joint density of (x, z), then we can write the observed log likelihood in the following way for arbitrary but fixed θ_0 in the parameter space

$$\log p(x|\theta) = \mathbb{E}[\log p(x, z|\theta)|\theta_0, x] - \mathbb{E}[\log p(z|x, \theta)|\theta_0, x] \tag{2}$$

Thus, to maximize the observed likelihood, we need only evaluate and maximize $Q(\theta|\theta_0, x) = \mathbb{E}[\log p(x, z|\theta)|\theta_0, x]$. That is, at iteration $t + 1$ we must perform the following E- and M-steps using current estimate $\hat{\theta}^{(t)}$.

$$\text{E-Step: Compute } Q(\theta|\hat{\theta}^{(t)}, x) = \mathbb{E}[\log p(x, z|\theta)|\hat{\theta}^{(t)}, x] \tag{3}$$

$$\text{M-Step: Compute } \hat{\theta}^{(t+1)} = ArgmaxQ(\theta|\hat{\theta}^{(t)}, x) \tag{4}$$

These steps are repeated until a sufficient convergence criterion has been met such as negligible difference in successive estimates. This is a well conditioned algorithm since the likelihood of successive estimates monotonically increases and can be shown to converge in probability to the maximum likelihood estimate. In the context of mixture model based clustering, given starting parameters, the E-step assigns the data to the most probable cluster and the M-step adjusts the parameters given the new assignments. Notice that constraining the structure of the covariance matrix creates a family of mixture models each with different behaviour. Therefore, to ensure an optimal fit, we evaluate from a family of models and choose the best performing structure. In particular, we use the Gaussian parsimonious clustering models (GPCM) and select via the Bayesian information criterion (BIC) which prioritizes performance as well as simplicity. An example of the GPCM family can be seen in Figure 5. Note that since we know that there are only two types of stars present in the data, we have that $G = 2$ a-priori making this a classification problem.

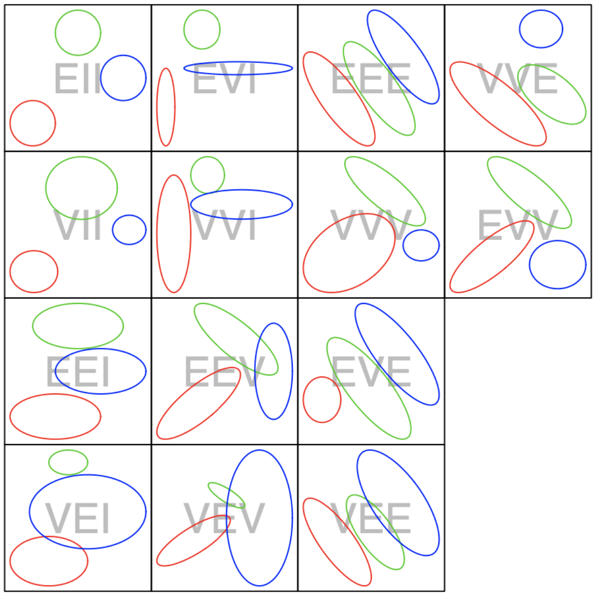


Figure 5: 2-dimensional GPCM example [6].

4 Results

4.1 Random Forest Results

As mentioned above, grid search was implemented to optimize the random forest over the hyper-parameter space consisting of 1 to 8 features and 100 to 500 trees. Figure 6 shows the results of this search. Notably, all models perform well with around 2% error, however, the optimal model is seen at the darkest node of the graph with 5 features and 200 trees. This implies that the unique IP fingerprints are much more important than the DM curves when it comes to classifying candidate stars which is not entirely surprising.

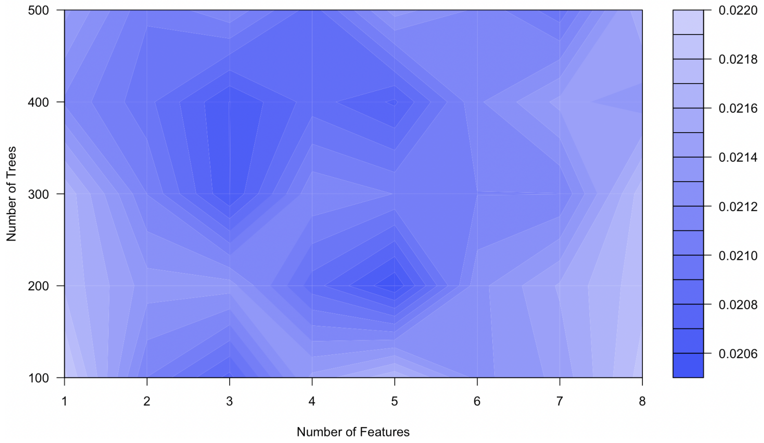


Figure 6: Grid search parameter space.

The final random forest was implemented using the optimal hyper-parameters and trained on the entire training set. The results are seen in Table 2.

Table 2: Random forest confusion matrix.

	Neutron	Pulsar	N
Neutron	5328	37	5365
Pulsar	76	465	541

The overall accuracy of the model is 98%. Although the accuracy is quite high, we note that it is comprised of 99% of the neutrons and 86% of the pulsars being accurately predicted respectively. Thus, the model is exceptional at noticing non-examples and moderately effective at noticing positive instances of pulsars.

4.2 Gaussian Mixture Model Results

Next, we implement the 14 Gaussian parsimonious clustering models. Two instances were run, one with k-means initialization and one with random soft initialization. The random initialization provided superior models in both accuracy and BIC. Table 3 shows the negative BIC of each model where the optimal structure is found to be VVV by far. Lastly, Table 4 shows the confusion matrix with an overall accuracy of 85% where the neutrons have an accuracy of 84% and the pulsars have an accuracy of 90%.

Table 3: Model BIC.

EVV	VEE	VVE	EVE	VVV	VEV	EEV	EEE	VVI	EVI	VEI	EEI	VII	EII
-29, 820.610	-101, 951.100	-122, 356.100	-112, 975.700	-5, 305.004	-18, 458.660	-39, 778.750	-227, 468.600	-182, 882.500	-221, 531.000	-241, 439.800	-346, 500.300	-300, 946.000	-352, 619.700

Table 4: Mixture model confusion matrix.

	Neutron	Pulsar	N
Neutron	13689	2570	16259
Pulsar	148	1491	1639

5 Conclusions

Neutron and pulsar stars were classified using random forests and Gaussian mixture models. The results show that random forests have a superior accuracy of 98% compared to the 84% from the mixture models. Although random forests have a higher overall accuracy, the mixture models are superior at correctly identifying pulsars. On the other hand, the random forests are superior at correctly identifying non-examples. We also note that the computational time of the mixture models was far superior to the random forests completing about 4 times faster. Overall, both techniques are useful in classifying pulsar stars and are both viable depending on the specific goals of the project.

References

[1] R.J. Lyon, et al., *Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach*: https://arxiv.org/pdf/1603.05166.pdf

[2] R.J. Lyon, *Why Are Pulsars So Hard To Find*: http://www.scienceguyrob.com/wp-content/uploads/2016/12/WhyArePulsarsHardToFind_Lyon_2016.pdf

[3] R.J. Lyon, *HTRU2 Data Set*: https://archive.ics.uci.edu/ml/datasets/HTRU2

[4] R.V. Hogg, J.W McKean, A.T. Craig, *Introduction to Mathematical Statistics: 8th Edition*

[5] T. Hastie, et al., *An Introduction to Statistical Learning*

[6] S. McNicholas, *Stats 4CI3 Lecture Notes: Mixture Model-Based Clustering 1-3*