**Data Science Final Project - Finding Pulsar Stars**

**Section B, Team 30**

**Team members:** Avani Bhargava, Chu-Ru (Ruth) Cheng, Roshni Shahani, Xinyan (Sarah) Wang, Yuchen (Vinnie) Zhang

**Introduction**

The data set analyzed is called “HTRU2” which describes a sample of pulsar candidates collected during the High Time Resolution Universe Survey[[1]](#footnote-0) [[2]](#footnote-1). Pulsars are a rare type of Neutron star that produce radio emission detectable on Earth. They are of considerable scientific interest as probes of space-time, the interstellar medium, and states of matter. Pulsar stars produce periodic pulsation signals, but the signals are hard to detect and can often be mixed up with cosmic noises. We are utilizing machine learning to conduct a classification task on pulsar star signal records to detect whether a signal is pure noise or a real pulsar star.

**Business Understanding**

Finding a pulsar is difficult. Not only is the radio wave emitted by a single pulse weak, but also the presence of radio frequency interference (RFI) and the background noise complicates the observation. Therefore, significant time and resources are needed in order to identify whether a detected radio wave pattern is emitted by a pulsar star. We attempt to build a screening model that can help NASA weed out candidates that are not pulsars so that they can save costs by better allocating their research budget. Mathematically, we are interested in E(value - cost| target). We also need to compare the impact of false positive E(cost | target) and false negative E(value | not target).

**Data Understanding**

The data set shared contains 17,898 records out of which 16,259 are spurious examples caused by RFI/noise, and 1,639 are real pulsar examples. Each

candidate is described by 8 continuous variables and a single class variable. The first four are simple statistics obtained from the integrated pulse profile (Mean of the integrated profile, Standard deviation of the integrated profile, Excess kurtosis of the integrated profile, Skewness of the integrated profile). This is an array of continuous variables that describe a longitude-resolved version of the signal that has been averaged in both time and frequency. The remaining four variables are obtained from the dispersion measure signal to noise ratio (DM-SNR) curve (Mean of the DM-SNR curve, Standard deviation of the DM-SNR curve, Excess kurtosis of the DM-SNR curve, Skewness of the DM-SNR curve).

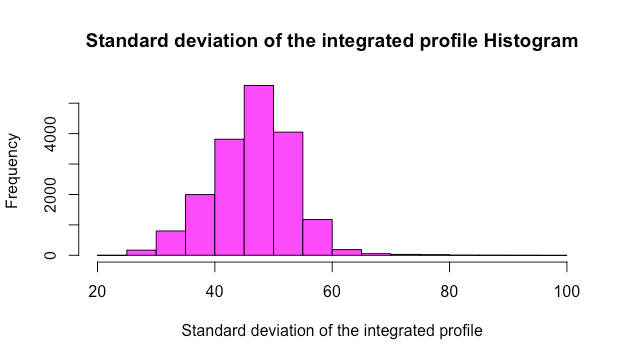
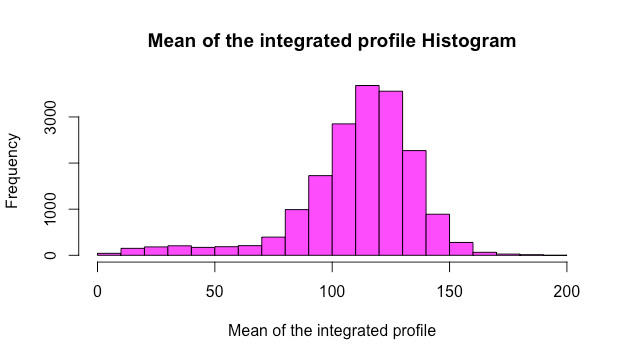
**Data Cleaning and Preparation**

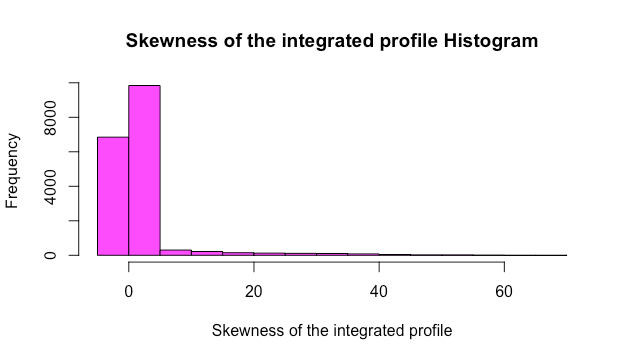
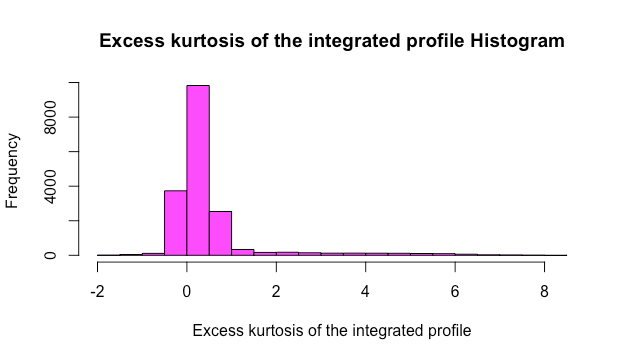
We explored the dataset for potential issues. No missing value was found. The class variable “target\_class” classifies the non-pulsar observations as 0 and real pulsar star observations as 1. We convert the “target\_class” variable into factors of ‘Yes’ and ‘No’ for recognition in R and simpler interpretation of our results signifying whether an observation is a pulsar star or not.

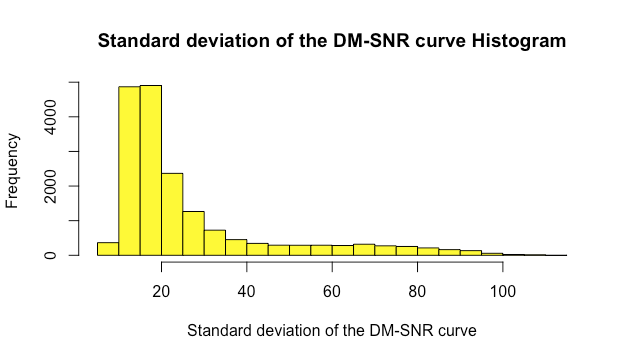
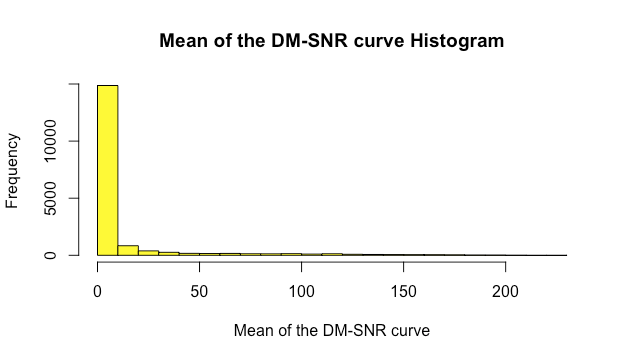
On further exploring the target variable, we observe that only ~9% of the radio emissions are detected as Pulsar Star emissions and ~91% are not. We then create a holdout sample to check whether the imbalance exists in both our ‘train’ and ‘test’ set and we find a similar result as the original dataset. To handle this extreme imbalance we use the SMOTE() function as a sampling method and we get a more balanced result of Pulsar Star emissions (~43%). SMOTE() function down-samples the large group and up-samples the small group to make the dataset more balanced.

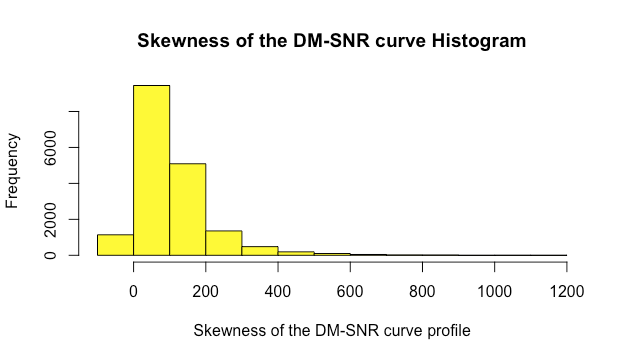
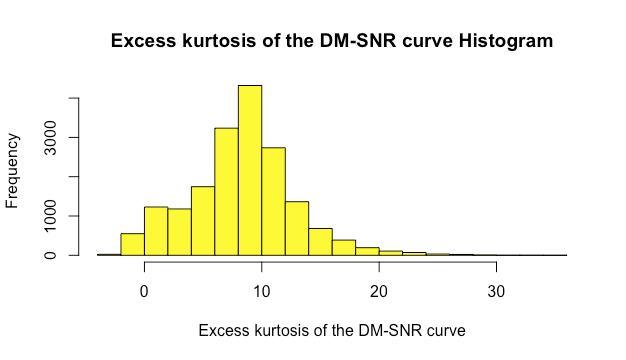
**Data Exploration**

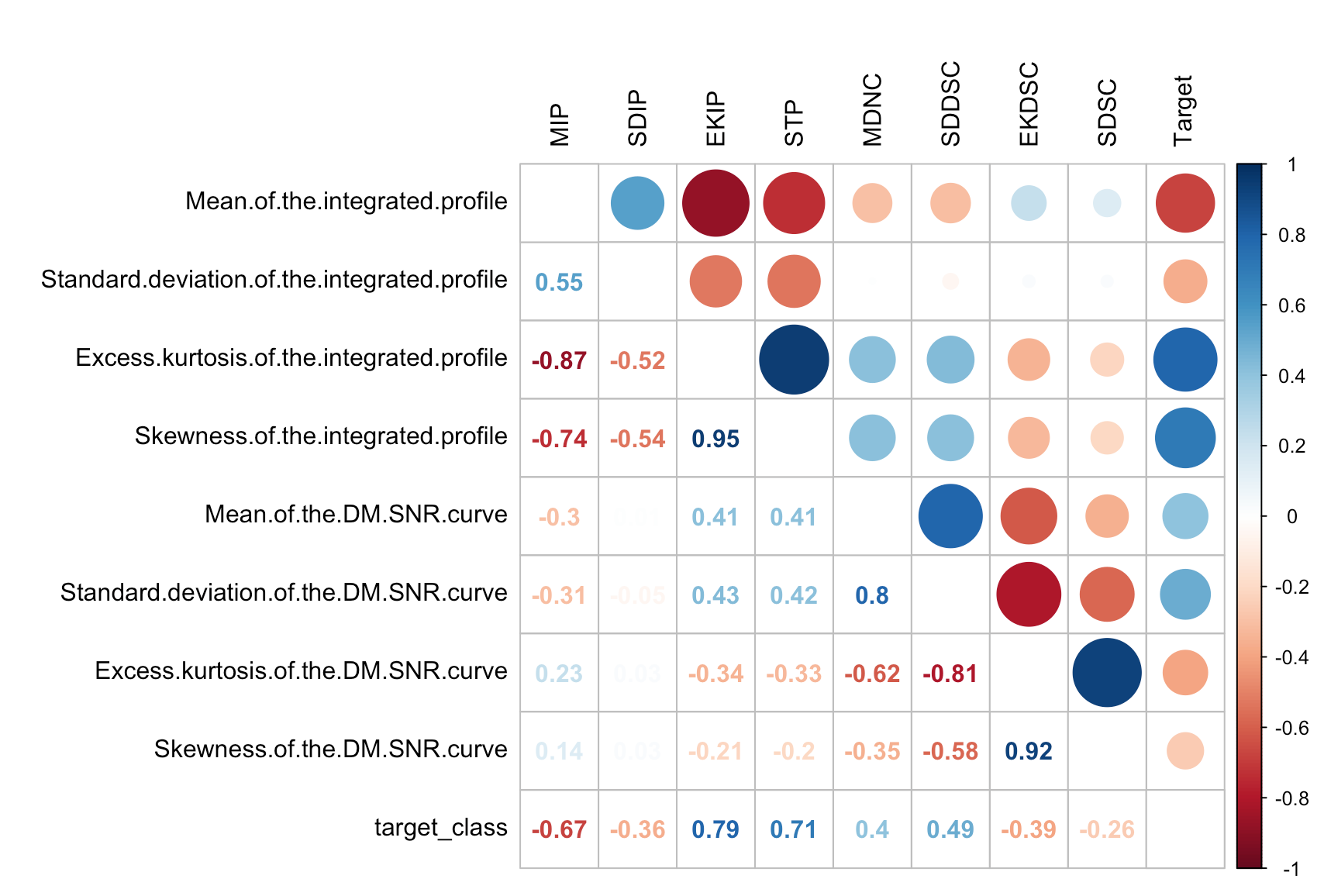
For exploratory analysis of the data, histograms for all the variables in our data set. We check for multicollinearity in our data using the correlation plot and observe that kurtosis tends to have high correlation with skewness. In further modeling we take special care to avoid multicollinearity.











Next, we performed unsupervised learning using k-means and PCA on our dataset to explore the data further and try to find any patterns and interpretations possible. Using information criteria as a standard, an 8-means clustering model was built for which two of the clusters seem to contain a majority of pulsar stars, with above 99% of observation within being real pulsar stars. These two clusters show similar values in their features. They have integrated profiles of high mean/standard deviation, and slightly low skewness/excess kurtosis. They also share DM.SNR curves of average mean, very high standard deviation, and very low skewness/excess kurtosis. These can be identifiable characteristics of real pulsars.

By plotting variance against the factors generated through PCA, we get three factors that seem explanatory. First factor is low mean, high skewness/kurtosis of integrated profile and high standard deviation, low kurtosis of DM.SNR, second factor is high mean, high standard deviation, low kurtosis of integrated profile and low skewness/kurtosis of DM.SNR and third factor is high mean and skewness of DM.SNR. The results of PCA does not show significant interpretability.

**Modeling**

In order to achieve the best accuracy in classifying the signal data, we utilized and compared 9 classification models in this supervised learning task. To illustrate the ability of the models to generalize on new, unlearned data, a 10-fold cross validation technique is applied to each model to generate an out-of-sample performance. The 9 models used are listed as following:

1. Logistic regression

A logistic regression with all the explanatory variables is built as a base model. Logistic regression is a widely used model in classification tasks and no specific feature selection is conducted here.

2. Logistic regression without highly correlated variables

For this logistic regression model, a cut-off of 0.9 is selected, and explanatory variables which have a correlation higher than the cut-off with each other are removed from the modeling process. This manual feature-selection approach tries to reduce multicollinearity.

3. Logistic regression with stepwise selection using AIC

This model utilizes an automated stepwise feature selection approach based on information criteria and removes features that are less informative. This approach is commonly adopted to reduce overfitting.

4. Logistic regression with interaction using Lasso

Interaction is used in this logistic regression model to address relationships between explanatory variables. L1 regularization (Lasso) is applied to conduct feature selection in order to reduce multicollinearity. Cross validation selects the best ƛ, which controls the amount of L1 penalty, for the model.

5. Decision tree

All 8 explanatory variables are used to build the decision tree model, with the value of complexity parameter being tuned by the function used. The complexity parameter (cp) is used to control the size of the decision tree and to select the optimal tree size. If the cost of adding another variable to the decision tree from the current node is above the value of cp, then tree building does not continue. Decision tree is very intuitive model and is easy to interpret, but have the tendency to overfit. The cross validation tries to reduce overfitting.

6. K-NN

All 8 explanatory variables are used to build the K-NN model, with the value of K being tuned by the function used. The optimal number of neighbors selected by then model is 17. K-NN is also a very intuitive model and it constantly evolves with increment in observations. However K-NN is also slow with large amount of observations and does not perform well when data is imbalanced. Due to the nature of the current business problem and data K-NN may not be the optimal choice.

7. Support Vector Machine (SVM)

The SVM model is built upon all 8 explanatory variables using a linear kernel. An advantage of SVM is the less impact of outliers, which can be typical in the aerospace setting. However SVM do require large amount of time to process and kernel selection can be tricky.

8. Random forest

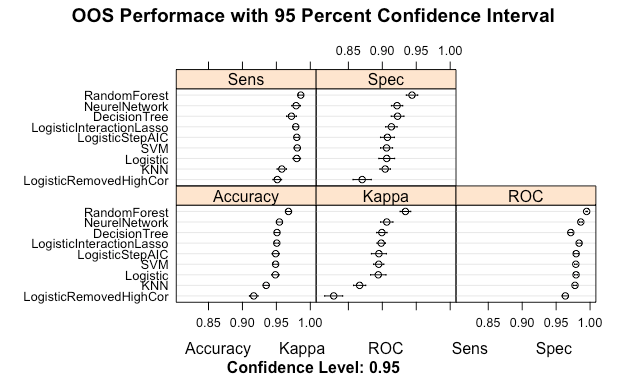
This ensemble model is applied to all 8 explanatory variables, with the number of features being randomly selected in each tree being tuned by the function used. The optimal number is a subset of 2 features. Random forest is important when dealing with multiple correlated features by providing a reliable feature importance estimate, but can be inherently less interpretable.

9. Neural network

A neural network model applies deep learning on the 8 explanatory variables, with the size -- the number of units in the hidden layer -- and decay -- the regularization parameter to avoid over-fitting -- being tuned by the function used. Neural network is a widely applicable model but is also the most computationally expensive and time-consuming. The model is also a black box and the amount of influence explanatory variables on the target variable is not interpretable.

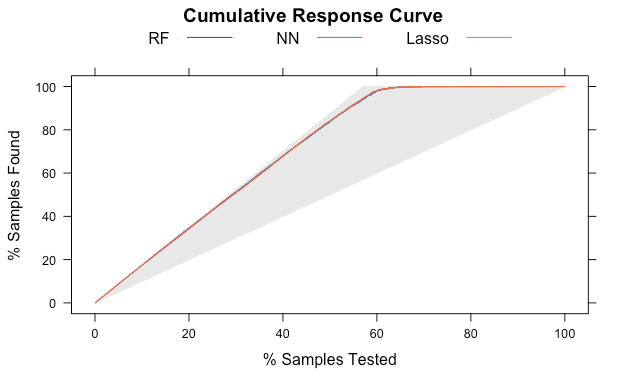
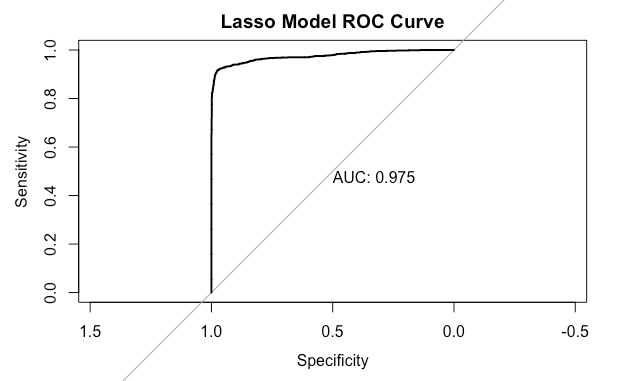
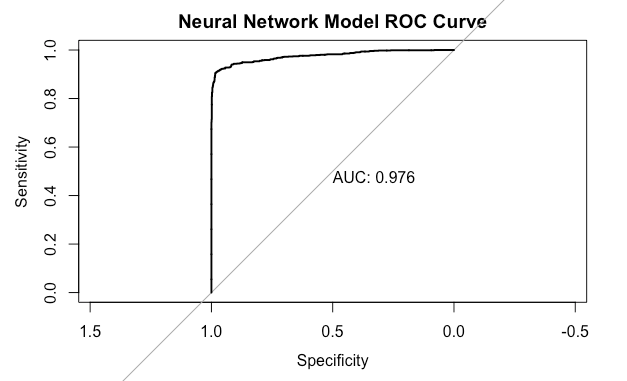
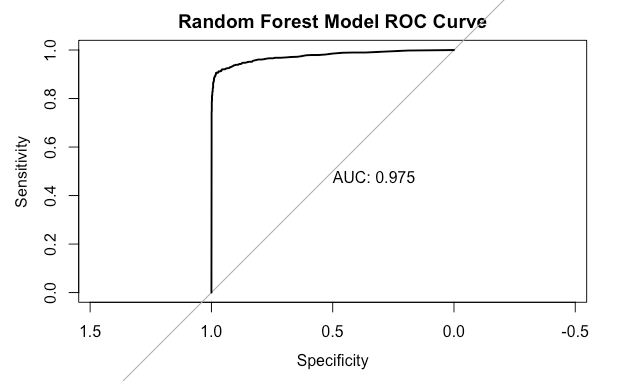
**Evaluation**

To evaluate the different supervised models, we measured their out-of-sample performance based primarily on out-of-sample area under the ROC curve (denoted ROC in the following graph). We choose AUC because it provides an aggregate measure of performance across all possible classification thresholds. Supplementary metrics are OOS sensitivity, OOS specificity, OOS accuracy, and OOS kappa statistic.



Random Forest ranked the highest but we decided to use the top three models, namely, Random Forest (AUC = 0.995), Neural Network (AUC = 0.986) and Logistic Interaction Lasso (AUC = 0.984).

The three models are then applied to the holdout test sample. By comparing different thresholds of 25%, 50% and 75%, for all three models the 50% threshold is selected as the optimal in accuracy. We further plotted ROC curves of model performance on the test set, and built a cumulative response curve, where the three models have very close performances well above randomization.



The random forest model, being the best in addressing multicollinearity, and not as computationally expensive as neural networks model is selected as the optimal model for the pulsar star classification task.

**Deployment**

Our project can be categorized into the aerospace industry. The astro-physical research process is time-consuming and costly. By applying machine learning and conducting classification on signal data, we can detect potential real candidates and eliminate obvious spurious candidates caused by noises in our model. This may further act as a preliminary test to new signal record to find plausible pulsar star candidates for conducting further research, thus saving time and budget, which are both crucial in astro-physical research.

However, the researcher should be aware that our model is not perfect. Since true pulsar stars are sparse in the dataset, more data might be needed in order to increase the accuracy of classification. In the model, we used 50% as the threshold based on accuracy. In reality, because we are dealing with a scientific environment, we are more tolerant to false positive cost E(cost | target) than we are to false negative misses where we lost E(value | not target). Thus researchers may be suggested to choose a lower threshold to reduce the false negative rate to reduce misclassification of true pulsar star. This means scientists and researchers may still need a lot of time to study on the filtered candidates from our model’s outputs.

Also, further studies may leverage astrophysicists’ professional opinions to improve and adjust our model. Prediction accuracy may be improved by combining human opinions and machine modeling together to detect new pulsar stars in the future.

1. [1] M. J. Keith et al., 'The High Time Resolution Universe Pulsar Survey - I. System Configuration and Initial Discoveries',2010, Monthly Notices of the Royal Astronomical Society, vol. 409, pp. 619-627. DOI: 10.1111/j.1365-2966.2010.17325.x [↑](#footnote-ref-0)
2. The Pulsar Stars dataset was collected by the School of Physics and Astronomy at the University of Manchester, and we found it on Kaggle’s dataset library [↑](#footnote-ref-1)