Pulsar Classification Using Automatic Machine Learning

Sam Johnson, Corwin Schmidt
Data Science Program, Indiana University, Bloomington, IN, USA
Computer Science Program, Indiana University, Bloomington, IN, USA

***Abstract*—Pulsars are of great scientific significance due to astronomers' ability to interpret their emissions as indications of extrasolar planets, stars, and more. By employing a multi-layer perceptron classifier, the radio emissions of pulsars can be identified amidst noise and Radio Frequency Interference (RFI) with an accuracy of over 98%. The ability to accurately discern the presence of pulsars can save astronomers time and money in their exploration of our galaxy and beyond.**

I. Background

When a sufficiently large star collapses following its supernova phase, it can form a super-dense mass known as a neutron star. Conservation of angular momentum means that such a mass typically spins extremely quickly, with some neutron stars rotating hundreds of times per second. Sometimes, a neutron star's electromagnetic field compresses in such a way that it forces thin streams of electromagnetic radiation from their magnetic poles [1]. Such a neutron star is known as a pulsar. As a pulsar spins about its axis of rotation, the beams of radiation it emits from its magnetic poles periodically face Earth, creating a "pulse" that can be registered as a radio frequency; hence the name, "pulsar."

Our ability to locate and verify pulsars is the basis for a great deal of important applications. While the consistency of the radio signals from pulsars can rival the atomic clock, the most useful information comes from the fluctuations in certain pulsars' emissions. Based on the changes in a pulsar's signal over time as it interacts with other objects and matter, astronomers can discover the presence of extrasolar planets and other stars. Additionally, astronomers can probe the interstellar medium, i.e. the matter (or lack thereof) between here and there, as well as space-time itself based on their readings of the pulsar's signal emission [1]. To summarize, pulsars allow us to understand areas of our universe that we might not be capable of viewing with telescopes for decades.

When it comes to the classification of pulsar radio signals, the issues of noise, RFI, and other signal sources present themselves. Without a suitable classification model, it may take an extended period of time to observe a radio signal and determine whether it is a pulsar or something else entirely. By applying machine learning (ML) techniques, we can dramatically reduce the amount of time needed to verify a radio source by classifying the emissions based on a relatively small amount of data captured over a relatively small period of time. Our goal in this project was to determine which model or models are most accurate and performant in this classification task.

II. Dataset

The HTRU2 dataset was used for this project. It is hosted by the UCI Machine Learning Repository and provided by Dr. Robert Lyon of the University of Manchester's School of Physics and Astronomy [2]. The data was collected in the High Time Resolution Universe Pulsar Survey [3]. Astronomers involved in this survey used the 13-beam multibeam receiver on the Parkes Radio Telescope to scan the entirety of the southern sky by dividing the Galactic latitudes into low, medium and high sectors.

The dataset itself includes a total of 17,898 examples, 16,259 of which are examples of noise and/or RFI, and 1,639 of which are verified pulsar emissions. We accessed these entries via the .csv file provided by Lyon to the UCI ML repository [2]. Each entry has 8 continuous variables and a single class variable. The first four continuous variables in the list below are statistics obtained from the folded profile of a radio emission source. Per the HTRU2 dataset, the folded profile is "an array of continuous variables that describe a longitude-resolved version of the signal that has been averaged in both time and frequency" [2]. The last four continuous variables are derived from the DM-SNR curve. The variables are as follows:

1) Mean of the integrated profile
2) Standard deviation of the integrated profile
3) Excess kurtosis of the integrated profile
4) Skewness of the integrated profile
5) Mean of the DM-SNR curve
6) Standard deviation of the DM-SNR curve
7) Excess kurtosis of the DM-SNR curve
8) Skewness of the DM-SNR curve
9) Class

III. Methodology

We selected the scikit-learn library in Python to complete the modeling for our classification task. Scikit-learn is an open-source software package that enables users to leverage data preprocessing, data analysis, ML modeling, and model evaluation in their programs. It is the most popular machine learning framework for data scientists in industry [4]. It is suitable for our classification task because it houses 30+ classification algorithms and supplies tools for visualizing model quality.

We began by importing all of classifiers in the scikit-learn package and our raw pulsar data from the UCI ML repository.

We then isolated the predictive variable that correlated the highest with our class label: the 'excess kurtosis of the integrated profile' variable. The data was divided into a training and test set by a 'stratified shuffle split,' which shuffles and splits the data while preserving the proportion of a chosen variable. We wanted to maintain the proportion of the variable with the highest correlation with our class label, so that our training set and test set were both accurate representations of the whole population.

We created a script that would perform training for each classifier, in turn, automatically. A framework in which data can be modeled with various machine learning algorithms at once is known as Automatic Machine Learning (Auto ML). Using Auto ML, we trained over 300 models using the cross-validation strategy. Cross-validation is the process of separating a training set into *k*-folds, then training the model on *k-1* set of folds. The remaining fold acts as a test set. This process indicates how well a trained model will perform on unseen data and can show the user if the model is victim to overfitting. We chose a value of ten for *k*, so ten individual models were created for each of the 30 classifiers. Due to the volume of models being trained, we did not tune the hyperparameters of the algorithms. 'Hyperparameters' are characteristics of each algorithm that may change performance of the model, based on the application. However, the default hyperparameter values provided by scikit-learn will yield reasonable results in training.

Following model training, we visualized the ten cross-validation runs as boxplots across three metrics. Algorithms were excluded from this final analysis if they threw errors in cross-validation. This could have been for a number of reasons, *e.g.* if the algorithm required extra data formatting, if the algorithm predicts multiple outputs, etc. The metrics we chose are accuracy, precision, and recall. Accuracy is the fraction of correct predictions to total predictions, precision is the proportion of correctly predicted positive instances to every actual positive instance, and recall is the fraction of correctly predicted positive instances out of every predicted positive instance [5]. The boxplots of those various metrics across the top 20 performing models are included in figure 1, figure 2, and figure 3, respectively.

IV. Results and Interpretation

Cross-validation training yields a model called the 'best estimator' that performed the best on the test or 'validation' fold. We used the best estimator from each algorithm to predict the class labels of the test set that we laid aside at the beginning of our project. The final prediction scores were recorded, and ranked based on their prediction accuracy. The best performing models in order of accuracy are the Multi-Layer Perceptron Classifier (0.9813 accuracy), the Histogram-based Gradient Boosting Classification Tree (0.9802), the Support Vector Classifier (0.9796), the Stochastic Gradient Descent Classifier (0.9796), and the Logistic Regression Cross-Validation Classifier (0.9793). Figure 4 displays the score of every estimator tested.

Each of the estimators exhibited an accuracy above 0.94, which would be consider strong in most settings. However, due to the significance of the application, a model is only suitable if it has a high precision and recall score as well. Precision is the ability of a model to not classify a negative instance as a positive instance. This is crucial in the task of classifying pulsars, as it would be a great loss of resources to study radio frequencies that were from an ostensible pulsar, but in reality

Fig. 1. Accuracy score across cross-validation training

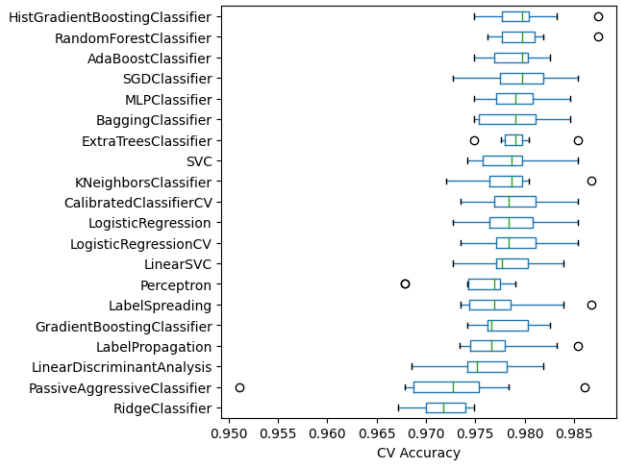


Fig. 2. Precision score across cross-validation training

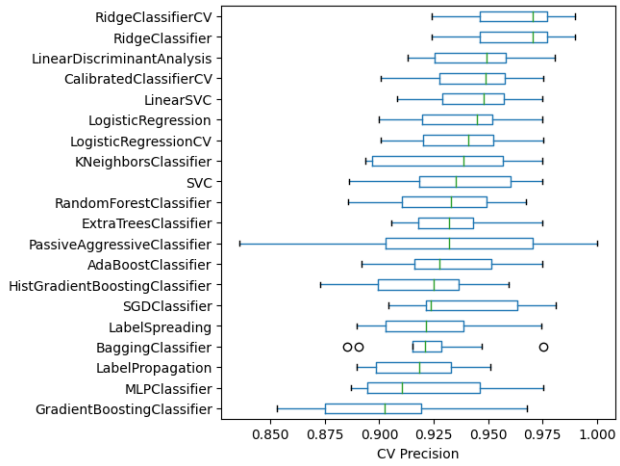
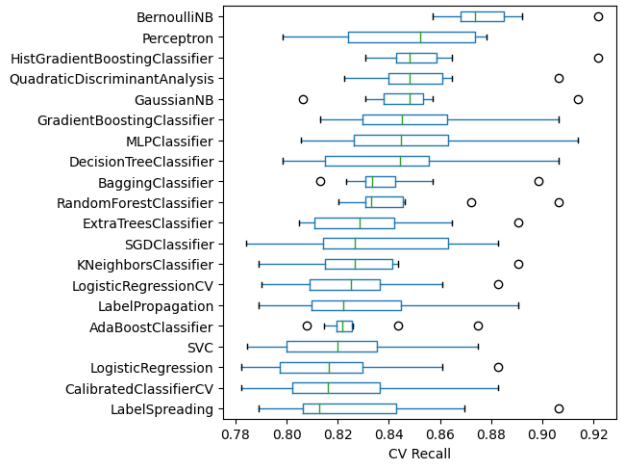


Fig. 3. Recall score across cross-validation training



were from another source of electromagnetic radiation.

The Multi-Layer Perceptron Classifier (MLP) performed exceptionally well at classifying unseen instances of the dataset. It exhibited the highest accuracy score of any classifier, along with strong precision (0.9444) and recall (0.8421) scores.

The Multi-Layer Perceptron Classifier is an artificial neural network (NN), an ML algorithm family modeled after the neurological functions that produce intelligence in humans. MLPs are composed of multiple layers of perceptrons. These perceptrons take a series of input variables and associated weights and computes the weighted sum of these variables. If the weighted sum is above a certain threshhold, the perceptron 'fires' like a neuron, sending the computation to the next layer. This continues until the data reaches the output layer, in which a continuous function is modeled. The MLP is trained by adjusting the weights of the input. This is performed via a process called backpropogation [6].

Fig. 4. Scores of each estimator on test set

	Accuracy	Precision	Recall
MLPClassifier	0.9813	0.9444	0.8421
HistGradientBoostingClassifier	0.9802	0.9375	0.8359
SVC	0.9796	0.9529	0.8142
SGDClassifier	0.9796	0.9562	0.8111
LogisticRegressionCV	0.9793	0.9527	0.8111
LogisticRegression	0.9793	0.9527	0.8111
BaggingClassifier	0.9791	0.9247	0.8359
CalibratedClassifierCV	0.9791	0.9526	0.808
RandomForestClassifier	0.9791	0.9247	0.8359
ExtraTreesClassifier	0.9791	0.9336	0.8266
LinearSVC	0.9785	0.9522	0.8019
KNeighborsClassifier	0.9782	0.9391	0.8111
PassiveAggressiveClassifier	0.9774	0.9764	0.7678
GradientBoostingClassifier	0.9771	0.8977	0.8421
AdaBoostClassifier	0.9768	0.9348	0.7988
LabelSpreading	0.9757	0.9275	0.7926
LabelPropagation	0.9749	0.9206	0.7895
LinearDiscriminantAnalysis	0.9732	0.9486	0.743
Perceptron	0.9723	0.8636	0.8235
RidgeClassifier	0.9704	0.9738	0.6904
RidgeClassifierCV	0.9704	0.9738	0.6904
DecisionTreeClassifier	0.9684	0.8261	0.8235
NearestCentroid	0.9676	0.8581	0.7678
ExtraTreeClassifier	0.967	0.8213	0.8111
QuadraticDiscriminantAnalysis	0.9665	0.7892	0.8576
RadiusNeighborsClassifier	0.9665	0.7892	0.8576
BernoulliNB	0.9522	0.6854	0.87
GaussianNB	0.943	0.64	0.8421

V. Key Takeaways

Among the models tested in this project, pulsars are most effectively classified using the Multi-Layer Perceptron Classifier. After extensive testing of multiple models compared against one another by a number of metrics, the MLP classifier had the best performance overall. With the extensive data provided by astronomical surveys of radio emissions, the model is capable of fine-tuning the recognition of pulsars as distinct from other sources of electromagnetic radiation.

By using the most accurate model possible to classify pulsars, astronomers can save time by quickly classifying examples of radio emissions, and can save money by avoiding expensive research conducted on misclassified emission sources. Accurately classified pulsars will contribute to our broader understanding of our galaxy and universe via their interactions (and their emissions' interactions) with planets, stars, the interstellar medium, and space-time.

To go further with this research, we could search the hyperparameter space of each algorithm. A fine-tuned model would likely produce higher accuracy than models trained using default hyperparameter values in scikit-learn. Additionally, it is possible that some of the variables from the dataset are less responsible for accurate classification, and could be either manipulated to improve the accuracy of the model or omitted entirely to improve time performance on much larger datasets. As sensors become more powerful, we will be able to scan wider sections of space at a much greater distance, discovering exponentially more pulsars and radio sources as time goes on. This might necessitate a faster method of processing the emission data. For now, and with this dataset in particular, we believe that our current model is suitably performant.

VI. Team Contributions

The group members added the following contributions to the group's analysis of pulsar classification methods: Sam Johnson located the dataset, wrote the Auto ML script used for modeling, generated the figures used in the report, completed sections III and IV of the report, finalized the document in IEEE style, and assisted with creating and formatting the slideshow presentation. Corwin Schmidt conducted background research on pulsars and their applications, created and formatted the slideshow, completed sections I, II, and V of the final report, and gathered sources that elaborate upon the most performant models explored in the analysis. Both Sam and Corwin organized the group and determined the group's task, and both took on additional responsibilities throughout the course of the project as prior members became unavailable.

References

[1] "Pulsars — Facts, Sound, Discovery, Information, History & Definition," The Nine Planets, Aug. 17, 2020. https://nineplanets.org/pulsars/ (accessed Dec. 13, 2022).

[2] R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, and J. D. Knowles, "Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach," Monthly Notices of the Royal Astronomical Society, vol. 459, no. 1, pp. 104–112, Feb. 2017, doi: 10.1093/mnras/stw656.

[3] M. J. Keith et al., "The High Time Resolution Universe Pulsar Survey – I. System configuration and initial discoveries," Monthly Notices of the Royal Astronomical Society, vol. 409, no. 2, pp. 619–627, Dec. 2010, doi: 10.1111/j.1365-2966.2010.17325.x.

[4] "State of Data Science and Machine Learning 2019," Kaggle. [Online]. Available: https://www.kaggle.com/kaggle-survey-2019. [Accessed: 20-Nov-2022].

[5] "3.3. metrics and scoring: Quantifying the quality of predictions," scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html. [Accessed: 20-Nov-2022].

[6] T. Menzies, E. Kocaguneli, L. Minku, F. Peters, and B. Turhan, "Chapter 24 - Using Goals in Model-Based Reasoning," in Sharing data and models in software engineering, Waltham, MA: Morgan Kaufmann, 2015, pp. 321–353.