HTRU Pulsar Classification

Eric Mai

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT
This project aims to explore various machine learning models and their efficacy when applied to the task of identifying pulsars, a historically laborious and tedious task. We use a dataset consisting of approximately 18,000 pulsar candidates from the High Time Resolution Universe (HTRU) survey, compiled by Robert Lyon, to train our various models on. It was initially planned that after developing several machine learning models to tackle the problem, the models will be compared for accuracy and efficiency. However, impressive results from the first couple of models introduced a pivot point to chose a single most suitable model and refine it until achieving a final test accuracy of 98.2%. It is possible that further tuning could produce even more accurate results, which future work will look to do.

Key words: Pulsar – Perceptron – SVM

1 INTRODUCTION

Traditionally, pulsar detection relies on manual candidate selection from large radio survey datasets, often using visual inspection and heuristic-based filtering techniques. This process is not only time-consuming but also subject to human bias, making it inefficient for handling the massive influx of data from modern radio telescopes. Machine learning, since it is already invading every other field of research, could be a potential solution to the tedious classification step of analyzing pulsars. In this project, I plan to explore various machine learning techniques, many of which I am not familiar with, to compare their efficiency in this particular classification problem.

Developing efficient and reliable methods to classify pulsars would have significant implications on astrophysics even if only purely to remove tedium from researchers who must manually classify data. In the age of "big data" astronomy where large amounts of resources are dedicated to processing massive amounts of data, anything to streamline the process would be a welcome improvement to quality of life. It could potentially free up time to perform more intense analysis, possibly accelerating scientific breakthroughs. More broadly, the application of machine learning in pulsar searches represents a step toward fully automated classification pipelines and more efficient processing of ever-increasing volumes of astronomical data.

2 BACKGROUND

Pulsars are highly magnetized, rapidly rotating neutron stars that emit beams of electromagnetic radiation, which can be detected as periodic pulses when aligned with Earth. They have a variety of uses in astrophysics due to their precise and stable rotational periods, which can be used to verify general relativity in extreme gravitational environments. Slight variations in their regular cadence could reveal the spacetime distortions caused by passing gravitational waves. Detecting and cataloging new pulsars is a high-priority task in observational astronomy, but identifying pulsars within vast astronomical datasets remains a challenging problem due to the overwhelming presence of random noise and interference in space.

3 PROBLEM DESCRIPTION

Astronomers frequently deal with large amounts of data, much of which is of no interest at all and full of cosmological noise. Pulsars, one item of particular interest, are rare, making identifying them within these large-scale radio survey data a challenging and time-consuming task. As modern radio telescopes generate increasingly large datasets, there is a growing need for an automated, scalable, and accurate classification pipeline. This project aims to address this problem by exploring and comparing various machine learning techniques to determine the most effective approach for pulsar candidate identification. By doing so, I hope to enhance the efficiency of pulsar searches, perhaps contributing to the discovery of new pulsars and advancing astrophysical research. If nothing else, I hope to learn more about machine learning implementations to eventually contribute to advancements in data heavy scientific fields.

4 RELATED WORKS

Serena Debesai, Carmen Gutierrez, and Nazli Ugur Koyluoglu from Stanford University published a paper in 2020 working with the same HTRU dataset (Debesai et al. (2020)). They compared methods of supervised and unsupervised learning and using modern machine learning techniques. The work in this paper was developed independently of finding their paper.

5 DATASET

The dataset we are working with was part of the HTRU survey, provided by Dr. Robert Lyon of the University of Manchester School of Physics and Astronomy in the United Kingdom (Lyon (2015)), and consists of approximately 18,000 pulsar candidates. It is given as a csv file with 8 columns for the feature vectors and the final column is the class. The feature vectors are described by the author as follows:

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 (folded 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 similarly obtained from the DM-SNR curve. These are summarised below:

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.

The dataset was split into training, validation, and testing datasets using scikit's 'test_train_split' method. The training data set is 70% of the entries, and validation and testing are each 15% of the dataset. The splits of the datasets were chosen randomly, but with logic to keep the relative number of true pulsars consistent across them.

5.1 Data Cleaning and Visualization

Before any models were trained, a preliminary step of data inspection and cleaning was performed. Since the dataset consisted solely of continuous numerical values with no missing entries, no imputation or removal of incomplete rows was required. However, due to the different statistical nature and scales of the features (for example, the mean of a distribution versus its skewness or kurtosis), it was necessary to normalize the data. To this end, the dataset was standardized using 'StandardScaler' from 'scikit-learn', transforming each feature to have zero mean and unit variance. This pre-processing step was done to assist in visualizing the dataset, described in further detail below.

To better understand the structure of the dataset and the separability of the classes, a two-dimensional visualization was generated using t-distributed Stochastic Neighbor Embedding (t-SNE), a non-linear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data. t-SNE is a nonlinear dimensionality reduction algorithm designed for embedding high-dimensional data in a lower-dimensional space (in our case, two dimensions) while preserving local structure. Introduced by van der Maaten & Hinton (2008), t-SNE converts similarities between data points into joint probabilities and minimizes the Kullback–Leibler divergence between these probabilities in the high-dimensional and low-dimensional space. Unlike linear methods such as Principal Component Analysis (PCA), t-SNE does not assume any linear relationships and is capable of capturing complex, non-linear manifold structures that may underlie the data.

The t-SNE plot (see Figure 1) revealed very little overlap between the two classes, but also displayed the non-linear nature of the dataset. A cluster corresponding to the true class, labeled with 1, appears distinct from the rest of the data points save for a few outliers. This qualitative insight helped to justify the use of non-linear models.

In addition to the t-SNE visualization, a feature correlation heatmap was created to investigate the dependencies between the input variables and identify the most informative features for the classification task. We used the 'heatmap' method in the data visualization library 'Seaborn' to generate the heatmap, which quantifies pairwise linear relationships using the Pearson correlation coefficient, r. Pearson correlation is defined by summing the product of the objects' differences from their object means, then dividing the sum by the product of the squared differences from the object means,

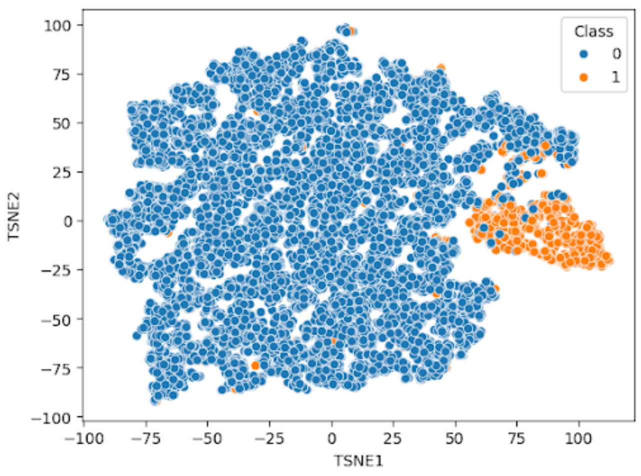
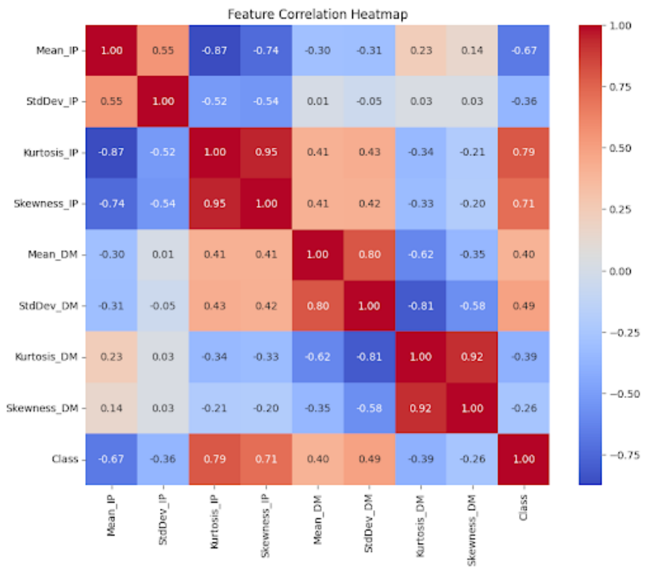


Figure 1. t-SNE visualization of the dataset in two dimensions. Points are color-coded by class label. A label of 1 represents the true pulsar class.



Feature Correlation Heatmap
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

	Mean_IP	StdDev_IP	Kurtosis_IP	Skewness_IP	Mean_DM	StdDev_DM	Kurtosis_DM	Skewness_DM	Class
Mean_IP	1.00	0.55	-0.87	-0.74	-0.30	-0.31	0.23	0.14	-0.67
StdDev_IP	0.55	1.00	-0.52	-0.54	0.01	-0.05	0.03	0.03	-0.36
Kurtosis_IP	-0.87	-0.52	1.00	0.95	0.41	0.43	-0.34	-0.21	0.79
Skewness_IP	-0.74	-0.54	0.95	1.00	0.41	0.42	-0.33	-0.20	0.71
Mean_DM	-0.30	0.01	0.41	0.41	1.00	0.80	-0.62	-0.35	0.40
StdDev_DM	-0.31	-0.05	0.43	0.42	0.80	1.00	-0.81	-0.58	0.49
Kurtosis_DM	0.23	0.03	-0.34	-0.33	-0.62	-0.81	1.00	0.92	-0.39
Skewness_DM	0.14	0.03	-0.21	-0.20	-0.35	-0.58	0.92	1.00	-0.26
Class	-0.67	-0.36	0.79	0.71	0.40	0.49	-0.39	-0.26	1.00

Figure 2. Feature correlation heatmap showing Pearson correlations between each variable and the class.

represented by the following equation. More about Pearson correlation can be found in Berman (2016), from where the equation is taken.

$$\frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2}\sqrt{\sum (y_i - \overline{y})^2}} \tag{1}$$

The heatmap (see Figure 2) revealed strong correlations between certain pairs of variables, with the strongest correlations being found in the excess kurtosis and skewness of the integrated profile—suggesting these features may be especially useful for identifying pulsar candidates. This observation would lead to a separate testing of a model exclusively trained on these variables.

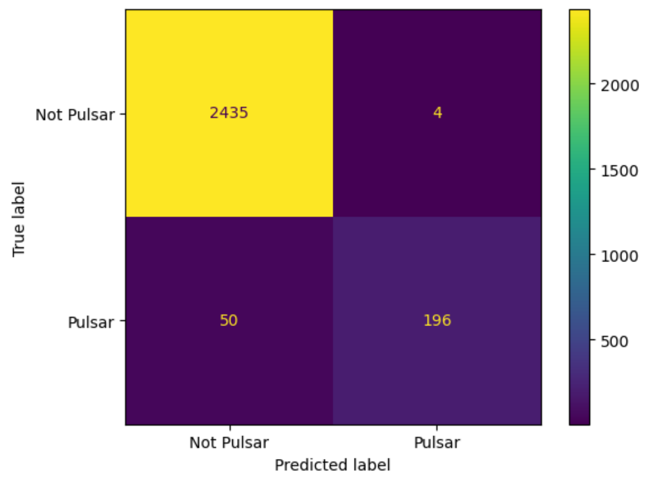


Figure 3. Confusion matrix for the Binary Perceptron model

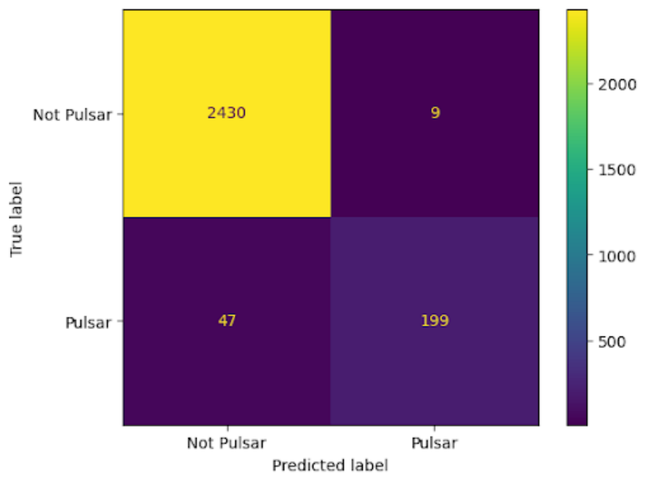


Figure 4. Confusion matrix for the Binary Perceptron model on the Cleaned Dataset

6 METHODS, OBSERVATIONS, SIMULATIONS ETC.

6.1 Binary Perceptron

The perceptron is one of the simplest types of artificial neural networks, so it was chosen to serve as a "baseline" model. It is a linear classifier that updates its weights iteratively based on misclassifications for a predetermined number of cycles.

Mathematically, the perceptron operates by computing a weighted sum of input features and passing the result through a step function to determine the predicted class:

$$y = \mathrm{sign}\left(\sum_{i=1}^{d} w_i x_i + b\right) \tag{2}$$

where:

- x_i represents the input features,
- w_i are the corresponding weights,
- b is the bias term,
- $\mathrm{sign}(\cdot)$ is the activation function, which outputs either $+1$ or -1 based on the sign of the weighted sum.

The perceptron learning rule updates the weights whenever the model misclassifies a training example. Given an input x with a true label y, if the predicted output is incorrect, the weight update follows:

$$w_i \leftarrow w_i + \eta \cdot (y - \hat{y}) \cdot x_i \tag{3}$$

$$b \leftarrow b + \eta \cdot (y - \hat{y}) \tag{4}$$

where η is the learning rate, and $\hat{y}$ is the predicted label. This iterative process continues over multiple epochs until the algorithm converges or reaches a stopping criterion.

The perceptron implemented was a relatively simple one and was only ran for 10 epochs. It ended with an accuracy rate of 97.9%. Its confusion matrix (see Figure 3) displays a proportionally large number of false negatives compared to false positives.

6.2 Binary Perceptron—Clean

The observation of integrated profile skewness and kurtosis having a high correlation with true pulsars prompted a separate experiment to be conducted. The initial developed perceptron was instead trained on a cleaned dataset consisting of only the ‘Kurtosis_IP‘ and ‘Skewness_IP‘ feature vectors. The cleaning of the dataset was simple; a section of the original dataset was taken using the ‘slice‘ operator for Pandas dataframes correlating to only the two relevant feature vectors. The perceptron model was then duplicated (to preserve the work done on the initial attempt) and modified to accept a smaller input matrix for its training data. It is potentially important to note that the testing/validation/training splits were generated again for this test and thus differ from the first attempt.

The perceptron was then run on its testing data split in the same manner as before, achieving an accuracy rate of 97.9%, differing from the first result in the fourth most significant digit. Oddly, this model did perform worse. This shockingly similar result, combined with the fact that randomness in the testing splits may contribute to slight variations in results, lead to the conclusion that the excluded feature vectors did not poison the results, and that future attempts should use the full dataset. The confusion matrix for this experiment (see Figure 4) shows slightly fewer false negatives, but they stil make up the majority of the incorrect labels.

6.3 SVM RBF

Due to initial observations during the data visualization phase that the data appeared to not be linearly separable, an SVM (Support Vector Machine) was chosen as an alternative model to try and achieve better results. SVMs are supervised learning models particularly well-suited for binary classification problems with high-dimensional feature spaces and non-linear data (Eskandar (2023)). Their primary objective is to construct an optimal decision boundary (hyperplane) that maximizes the margin between the two classes in feature space.

In the case of linearly non-separable data, SVMs employ a technique known as the kernel trick, which implicitly maps the original feature vectors into a higher-dimensional space where a linear separator may exist. The RBF kernel is among the most widely used due to its ability to model non-linear relationships. It is defined as:

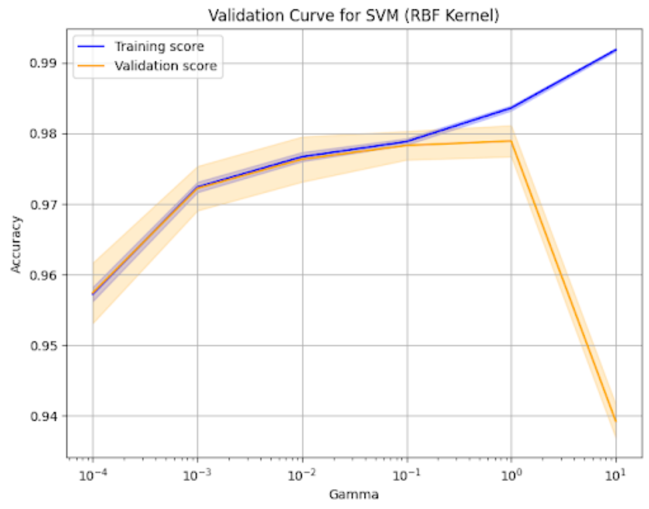


Figure 5. Validation curve for the SVM RBF

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right) \quad (5)$$

where γ is a hyperparameter controlling the influence of a single training example. A small γ value implies a large radius for the influence of each support vector, resulting in smoother decision boundaries, whereas a large γ tends to overfit by creating highly localized decision regions.

The SVM was implemented using scikit-learn's 'SVC' class with 'kernel='rbf''. Hyperparameters C (the regularization parameter) and γ were tuned using a validation curve, wherein the model was trained multiple times with varying values of one parameter while holding the other constant. This allowed us to identify the balance between model complexity and generalization performance. The validation curve is presented in Figure 5. As shown in the graph, the standard γ value of 0.1 was shown to be the most effective, and thus was selected for the test.

The SVM model yielded strong classification performance, but did not significantly outperform the initial binary perceptrons. While the theory was that the ability to construct complex, non-linear decision boundaries would make it especially effective in capturing the nuanced separation between pulsar and non-pulsar candidates, the model still finished with a testing accuracy of 98.2%. This result is still better, and given the small amount of room for improvement, this was considered the best model.

6.4 Alternative Models—k-Nearest Neighbors, Decision Tree, Random Forest

The impressive results from the models so far introduced the question as to whether continuing to experiment with various machine learning models would produce significant improvement in results. To approach this without having to develop extensive alternative models, a short experiment was designed using tools provided by scikit-learn. A dataset with a similar visual separation between the classes to the one observed in the dataset (see Figure 6) was constructed using the 'make_moons' method. The goal was to construct a dataset that was not immediately linear separable, but still had distinct classes.

A collection of basic models, including k-Nearest Neighbors, SVM, Decision Tree, and Random Forest, were then run on the dataset and then given an accuracy score. The results of the experiment were compiled into a graph (see Figure 7) for easy comparison. Based off the simple results, it seemed inefficient to continue persuing alternative models, since none were able to outperform the previous SVM RBF.

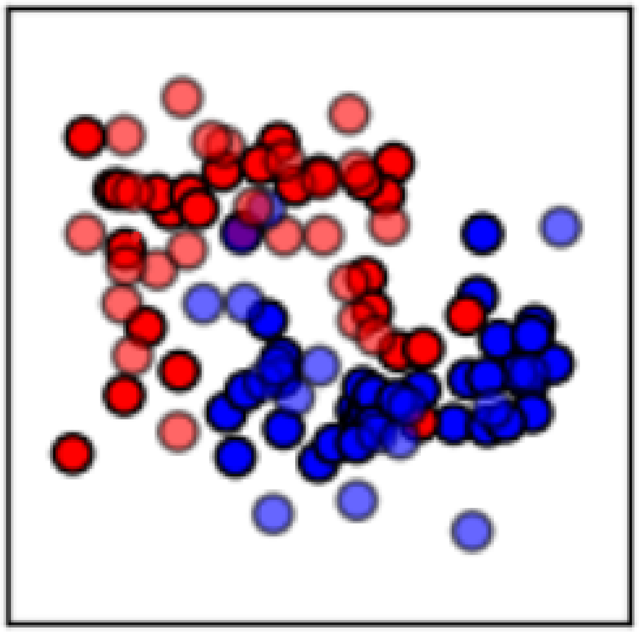


Figure 6. Dummy dataset constructed with make_moons

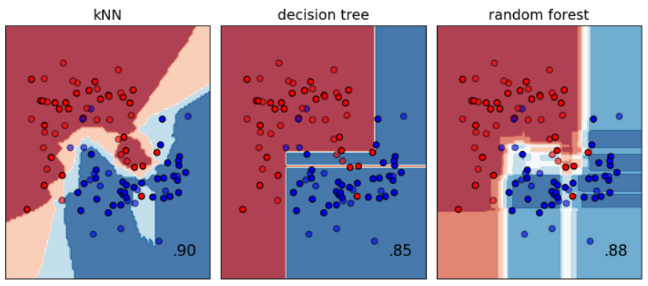


Figure 7. Compilation of the trials for alternate models. Accuracy scores are displayed in the bottom right corner of individual graphs.

7 CONCLUSIONS

This study has undertaken a comprehensive investigation into the application of machine learning techniques to the problem of pulsar candidate classification, using a dataset derived from the High Time Resolution Universe (HTRU) survey. The dataset represents a real-world classification challenge in radio astronomy, where the need for accurate and automated pulsar detection continues to grow with the increasing scale of survey data.

Our initial exploration involved thorough pre-processing and visualization of the dataset. The application of t-distributed Stochastic Neighbor Embedding (t-SNE) facilitated the projection of the eight-dimensional feature space into a two-dimensional representation, enabling qualitative assessment of class separability. Complementary to this, a feature correlation heatmap revealed strong relationships between specific statistical properties—most notably, the skewness and kurtosis of the integrated pulse profile—and the target classification. These exploratory techniques provided key insights into the structure of the data and informed subsequent modeling decisions.

We first implemented a classical Perceptron algorithm as a baseline linear classifier. Despite its simplicity, the model demonstrated a high degree of accuracy, achieving approximately 97% on the held-out test set. This model performed similarly with a cleaned dataset, leading to a reversion of the data cleaning step as it was determined that no feature vectors were poisoning the data.

Subsequently, we employed a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel to introduce non-linear decision boundaries. The RBF kernel enabled the SVM to project the data into a higher-dimensional feature space, allowing for the capture of more intricate relationships between features. A validation curve was utilized to investigate the sensitivity of the model to the γ hyperparameter, guiding optimal model selection while mitigating the risk of overfitting.

The outcomes of this investigation reaffirm the efficacy of machine learning methodologies in pulsar candidate classification. Both the linear and non-linear models achieved strong performance metrics, suggesting that the intrinsic structure of the dataset is conducive to algorithmic learning. The use of transparent, interpretable models such as the Perceptron also supports the pursuit of explainable machine learning within the scientific context.

In conclusion, this project has demonstrated the applicability and promise of machine learning in the domain of radio pulsar classification, providing a foundation for continued development and refinement of predictive models in the context of large-scale astronomical surveys.

ACKNOWLEDGEMENTS

The author would like to thank Professor Jay Lim for his support throughout the semester, and the Yale Computer Science Department.

DATA AVAILABILITY

The HTRU2 pulsar dataset is provided by Dr. Robert Lyon of the University of Manchester School of Physics and Astronomy in the United Kingdom and can be found at the following link.

All referenced papers can be reached thorugh their citations.

REFERENCES

Berman J. J., 2016, in Berman J. J., ed., , Data Simplification. Morgan Kaufmann, Boston, pp 135–187, doi:https://doi.org/10.1016/B978-0-12-803781-2.00004-7, `https://www.sciencedirect.com/science/article/pii/B9780128037812000047`

Debesai S., Gutierrez C., Koyluoglu N. U., 2020, PhD thesis, University of Stanford

Eskandar S., 2023, Introduction to RBF SVM: A Powerful Machine Learning Algorithm for Non-Linear Data

Lyon R., 2015, HTRU2, UCI Machine Learning Repository

van der Maaten L., Hinton G., 2008, Journal of Machine Learning Research 9

This paper has been typeset from a TEX/LATEX file prepared by the author.