Predicting Pulsars

Stefan Luncanu

August 2024

1 Motivation of choosing this topic

I chose as my topic a pulsar prediction algorithm because of my love for astrophysics and physics. This project ties in with my academic interests and an upcoming interview at Romania's largest laser facility [2], where I'll be working with experimental physics data. It's a fantastic opportunity to deepen my understanding of the universe and get ready for the practical aspects of my career, merging theory with real-world applications.

In this project, I have incorporated most of the algorithms and techniques presented in the course to analyze the pulsar dataset. Each algorithm was applied to compare their effectiveness and accuracy in identifying genuine pulsar signals from spurious noise. This comparative approach allowed me to evaluate the strengths and limitations of each method under different conditions and configurations, providing a comprehensive understanding of which techniques are most suitable for this type of data analysis.

2 Introduction about the topic

A pulsar, presented in Figure 1, is a rapidly spinning neutron star with strong magnetic fields, emitting beams of electromagnetic radiation from its poles. This radiation is only detectable when one of these beams points towards Earth, making the pulsar appear to pulse. Pulsars are extremely dense, spin regularly, and have very precise intervals between pulses, which can last from milliseconds to seconds. They are also considered potential sources of ultra-high-energy cosmic rays.

The challenge in pulsar detection is distinguishing genuine signals from spurious ones caused by "noise". To address this, machine learning, particularly binary classification systems, is applied to label pulsar candidates quickly and efficiently.

2.1 Dataset

The raw observational data for this study was gathered by the High Time Resolution Universe Collaboration at Parkes Observatory, funded by the Common-

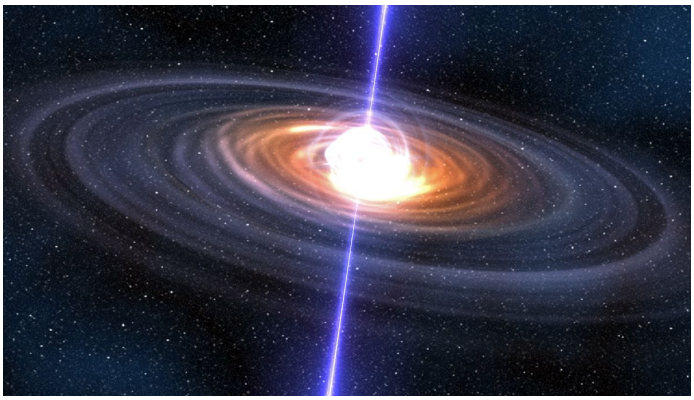

Figure 1: Rendered image of a pulsar [1]

wealth of Australia and managed by CSIRO. The dataset, known as HTRU2 [3], consists of pulsar candidates identified during the High Time Resolution Universe Survey (South).

The provided dataset contains 16,259 false examples and 1,639 real pulsar examples, all validated by human annotators and characterized by 8 continuous features derived from the analysis of pulsar signals and a class label.

The features result from a longitude-resolved version of the signal that has been averaged in both time and frequency, using traditional methods of pulsar candidate analysis. These methods include examining the integrated pulse profile and analyzing the Dispersion Measure/Signal-to-Noise Ratio (DM/SNR) curve. Here's a breakdown of the features:

- Mean of the integrated profile - This is the average value of the integrated pulse profile.
- Standard deviation of the integrated profile - Measures the variability of the integrated pulse profile.
- Excess kurtosis of the integrated profile - Indicates the peakedness of the integrated pulse profile.
- Skewness of the integrated profile - Captures the asymmetry in the integrated pulse profile.
- Mean of the DM-SNR curve - Average value of the DM-SNR curve.
- Standard deviation of the DM-SNR curve - Indicates the variability in the DM-SNR curve.
- Excess kurtosis of the DM-SNR curve - Measures the peakedness of the DM-SNR curve.

- Skewness of the DM-SNR curve - Captures the asymmetry in the DM-SNR curve.

The dataset is structured in a CSV format where each row corresponds to a pulsar candidate. The last column holds the class label (1 for real pulsars and 0 otherwise), and the remaining eight columns contain the values for the features listed above. This structured approach allows for efficient analysis and machine learning model training to distinguish between genuine pulsar signals and noise/spurious signals.

3 Code Walk-through

This code will take more than 10 minutes to run due to the implementation of multiple methods. The comprehensive nature of these methods ensures a thorough analysis, but it also extends the runtime.

3.1 Importing the libraries and reading the data

I've chosen to place nearly all the libraries used at the start of the '.ipynb' file, as this aligns with my usual practice.

The columns didn't have a name in the csv file so i have put them the corresponding names from the dataset website [3] for a more organised approach.

3.2 Feature selection

This dataset has only 8 features. I have decided to do a correlation matrix, presented in Figure 2, of them to see if there are any features that need to be eliminated due to strong correlations to each other or they have little to no correlation to the target itself. I decided to eliminate the "'Excess kurtosis DM SNR" and "Excess kurtosis integrated profile" features due to high correlations with other features.

3.3 Splitting the data and plotting the relations between the features and label

I splitted the features (x) from the label (y) and i plotted each pair (x, y) to check for outliers.

I must add that there are many outliers that could be removed from the dataset because they are not within IQR. As a future plan, i will improve the accuracy metrics of this program by finding which candidates to eliminate from the dataset.

3.4 Scaling

I have used RobustScaler and MinMaxScaler in the final implementation of this program. This pair gave me the best accuracy metrics for the dataset used.

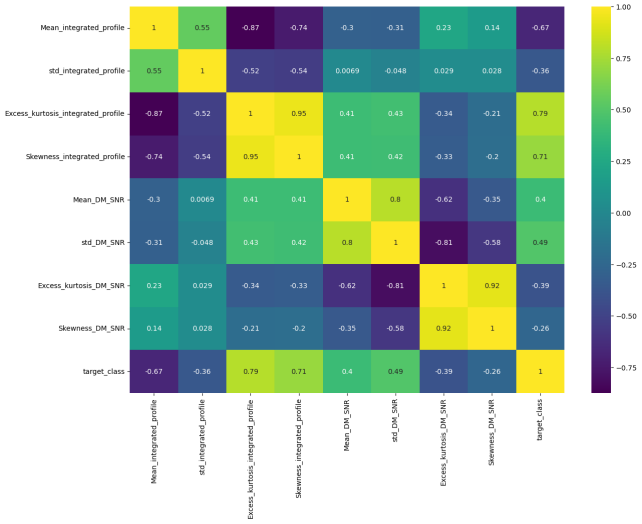

Figure 2: Correlation Matrix

The Robust Scaler is effective in reducing the impact of outliers by using the median and interquartile range for scaling, which is crucial given the presence of noise and anomalies in pulsar signal data. Following this, the MinMax Scaler can normalize these adjusted values to a specific range, 0 to 1, enhancing the performance of this ML algorithm. This two-step scaling process ensures that the data is both outlier-resistant and optimally scaled for subsequent analysis.

3.5 Splitting the data for training and testing

I've used a 75/25 training to testing ratio for this code.

3.6 Modeling

As mentioned in the first section, I've chosen to use multiple classifiers in this program to reinforce the skills I acquired during the course.

- - LogisticRegression
- - DecisionTreeClassifier
- - RandomForestClassifier
- - KNeighborsClassifier
- - XGBClassifier
- - VotingClassifier (hard , soft)

- - SVM

I also incorporated the typical performance metrics used for evaluating classification models:

- - Confusion Matrix
- - Precision
- - Recall
- - F1-Score
- - AUC ROC Curve

3.6.1 PrintReportFunction

Using ChatGPT, I developed a print report function that simplifies our workflow by removing the need to repeatedly paste the same code for each model. This function not only makes our tasks more straightforward but also streamlines the entire analysis process. Additionally, every time this function is run, it records the model and its performance metrics in a dictionary named "compa," further facilitating the comparison of different algorithms. This enhancement significantly eases the evaluation and comparison stages of our project.

3.6.2 Models used

Now, i started coding each model and print the accuracy metrics, the correlation matrix, AUC and ROC Curve for each one.

- Logistic Regression: using GridSearch for finding the best parameters
- Decision Tree: using GridSearch for finding the best parameters
- Random Forest: without GridSearch because of high computation time
- KNN: i applied the in-class algorithm of finding the optimal k number of neighbors
- XGBoost: using GridSearch for finding the best parameters
- HardVoting: applied the classifier with the following basic (not tuned) estimators: LogisticRegression, DecisionTree, RandomForest, AdaBoost and XGB
- SoftVoting: applied the classifier with the following basic (not tuned) estimators: LogisticRegression, DecisionTree, RandomForest, AdaBoost and XGB
- SVM: using GridSearch for finding the best parameters; using polynomial kernel trick

3.7 Comparision between the models and analysis of the results

Using the dictionary "compa" that stored the models and their performance metrics,I have done a comparision of all the models described earlier. The plot is presented in the Figure 3.

In the dataset, all the models demonstrate a pattern where accuracy is the highest metric, followed by precision, then the F1 score, and finally recall as the lowest. This occurs because accuracy benefits from the large number of true negatives in common datasets, boosting its value. Precision, which only looks at positive predictions, is naturally lower without the boost from true negatives. The F1 score, which combines precision and recall, is lower still because it's dragged down by the typically lower recall. Recall is the lowest because it measures how many actual positives are correctly identified, and a lower recall indicates that the model misses many positive cases.

This sequence reflects a trend where the models are efficient at classifying negatives but less effective at consistently identifying all positives.

The performance metrics chart for various machine learning models illustrates that each model excels in different areas—accuracy, recall, precision, and F1 score—without a single model consistently outperforming the others across all metrics.

This variety in strengths suggests that choosing the best model depends on the specific needs of the application, highlighting the absence of a one-size-fits-all solution among these options.

Although there is no clear winner in this "battle", if i would have to choose one, i would choose XGBClassifier for this dataset because of its high recall,f1 and decent accuracy and precision.

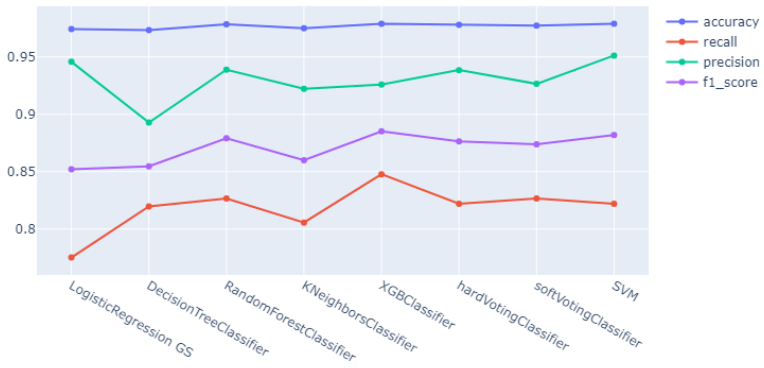

Figure 3: Comparision between the models

4 User interface

I have created a user-friendly interface using TKinter, designed for scientists to input data from detectors and receive immediate predictions (0 means no pulsar, 1 means pulsar) from various models. For example in the screenshot shown in Figure 4, all of the classifiers predict the data to not correspond to a pulsar. This tool simplifies data analysis, providing instant feedback directly through a graphical interface, making it accessible and efficient for users.

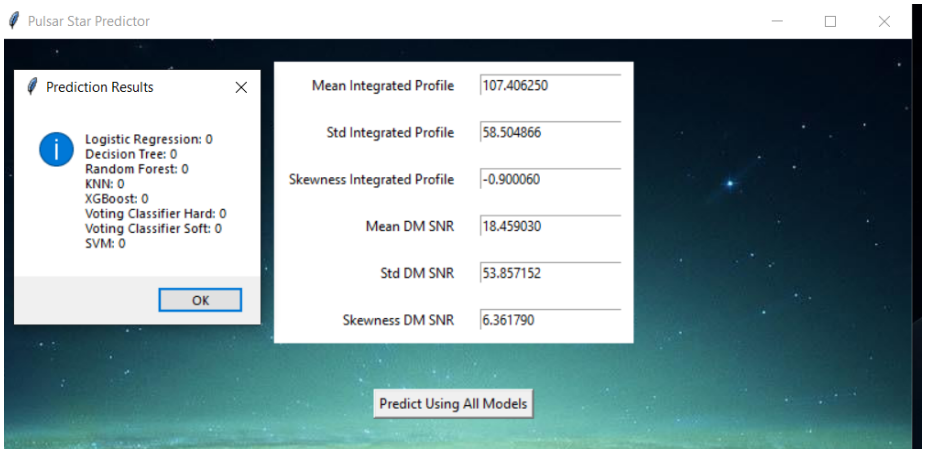

Figure 4: Prediction Interface

5 Reflection

Reflecting on the machine learning course I recently completed, I feel immensely grateful for having chosen to enroll. Before this course, I was uncertain about my career direction, but now, I am strongly inclined towards a future in AI. The experience has not only solidified my academic interests but also my appreciation for South Korea, where I am now considering pursuing a master's or PhD.

The course provided a comprehensive exploration of key machine learning algorithms such as K-Nearest Neighbors, K-Means, Support Vector Machines, Decision Trees and Random Forests. Learning to apply these techniques has equipped me with a robust foundation for addressing real-world data challenges effectively. Moreover, it has inspired me to delve deeper into more complex areas of the field.

The materials provided for the course were incredibly helpful and played a crucial role in our understanding of the subject matter. They offered clear explanations and practical insights, which greatly assisted me in studying new concepts.

Additionally, the course sparked my interest in researching and understand-

ing academic papers, keeping me updated on the latest advancements and interesting topics.

In conclusion, I am deeply thankful to my professor and colleagues for their collaboration throughout this course. Their support and insights greatly enhanced my learning experience, making it not only educational but also incredibly enjoyable.

hyperref

References

[1] https://assets.iflscience.com/assets/articleNo/64624/aImg/56994/artist-impression-of-a-pulsar-image-credit-michaeltaylor-shutterstock-com-meta.jpgArtist impression of a pulsar. Image credit: Michael Taylor / Shutterstock.com.

[2] https://www.eli-np.ro/Extreme Light Infrastructure - Nuclear Physics (ELI-NP).

[3] https://archive.ics.uci.edu/dataset/372/htru2HTRU2 Dataset at UCI Machine Learning Repository.

[4] R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, 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. 1104–1123, DOI: 10.1093/mnras/stw656.

[5] Debesai, Serena and Carmen Gutierrez. "Application of machine learning methods to identify and categorize radio pulsar signal candidates," 2020.