Predicting Pulsar Stars
Data Analysis and Preparation

Duc Ngo - CS4300

February 24, 2020

Contents

1 Introduction

What is a Pulsar? From Earth, pulsars frequently look like glinting stars, and they appear to sparkle with a normal cadence. Nonetheless, the light from pulsars doesn't sparkle or beat, and these celestial bodies are not stars. Pulsars are round, minimal objects that are about the size of a huge city however contain more mass than the sun. [1]

Why are they so fascinating? Researchers are utilizing pulsars to analyze the extreme states of matter, look for planets past Earth's nearby planetary group and measure cosmic distances. Pulsars likewise could assist researchers with finding gravitational waves, which could guide the way to energetic cosmic events like collisions between supermassive black holes. We study pulsars because they are so oddly fascinating and so different from anything we experience here on Earth. They are by definition is an extreme science. [1]

Motivation Artificial intelligence could be the perfect tool for exploring the Universe since finding a specific celestial object is slow and tedious work. With this in mind, the motivation of this project is to try to write supervised learning algorithms to predict the class of pulsars with logistic regression. Logistic regression is a measurement of the relationship between the categorical dependent variable and one or more independent variables.

Google Colab `https://colab.research.google.com/drive/1yscRn_wJqhPNDB4SjhWdvAQ3PSFghxzr`

2 Dataset

The "Predicting a Pulsar Star" dataset was obtained from the Kaggle Data Science [2][3]. The pulsar candidate data were obtained during the High Time Resolution Universe (HTRU) survey. It contains 16,259 apocryphal (negative) examples caused by RFI/noise and 1,639 real pulsars (positive) examples which totals to 17,898 samples in the dataset. These samples have all been checked by human annotators. Each row lists the variables first that described by 8 continuous variables, and a single class label as the final entry. The class labels used are 0 (negative) and 1 (positive). The input features are for the following fields:

1. Mean of the integrated profile.
2. The standard deviation of the integrated profile.
3. Excess kurtosis of the integrated profile.
4. The skewness of the integrated profile.
5. Mean of the DM-SNR curve.
6. The standard deviation of the DM-SNR curve.
7. Excess kurtosis of the DM-SNR curve.
8. The skewness of the DM-SNR curve.
9. Class

2.1 Visualization of the distribution of each input features

The histogram plot of each info highlights indicating their most extreme and least value as well as how they are distributed can be found in the pictures given underneath.

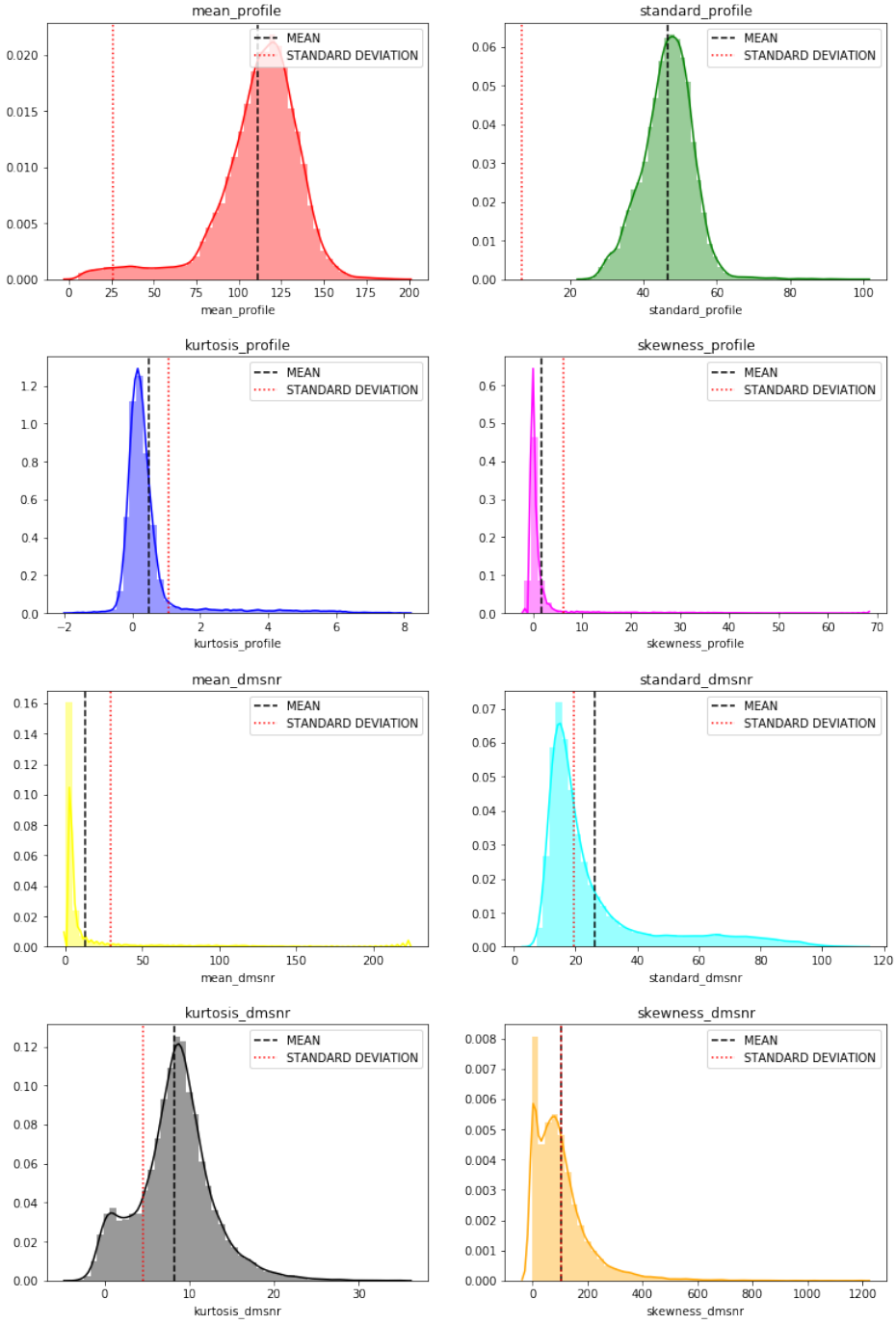

Figure 1: Input Data Distribution Histograms

	mean_profile	standard_profile	kurtosis_profile	skewness_profile	mean_dmsnr	standard_dmsnr	kurtosis_dmsnr	skewness_dmsnr	target
count	17898.000000	17898.000000	17898.000000	17898.000000	17898.000000	17898.000000	17898.000000	17898.000000	17898.000000
mean	111.079968	46.549532	0.477857	1.770279	12.614400	26.326515	8.303556	104.857709	0.091574
std	25.652935	6.843189	1.064040	6.167913	29.472897	19.470572	4.506092	106.514540	0.288432
min	5.812500	24.772042	-1.876011	-1.791886	0.213211	7.370432	-3.139270	-1.976976	0.000000
25%	100.929688	42.376018	0.027098	-0.188572	1.923077	14.437332	5.781506	34.960504	0.000000
50%	115.078125	46.947479	0.223240	0.198710	2.801839	18.461316	8.433515	83.064556	0.000000
75%	127.085938	51.023202	0.473325	0.927783	5.464256	28.428104	10.702959	139.309331	0.000000
max	192.617188	98.778911	8.069522	68.101622	223.392140	110.642211	34.539844	1191.000837	1.000000

Figure 2: Input Feature Statistics

2.2 Distribution of the output labels

Notice that the data is imbalanced and may need to be resampled (such as oversampling, undersampling, or generate synthetic samples), but for now it is acceptable.

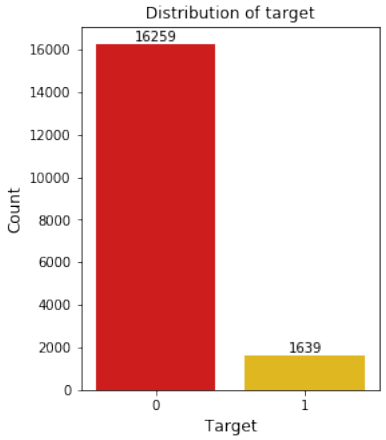

Figure 3: Output Data Distribution Histogram

3 Data Processing

3.1 Data Normalization

Before data mining itself, data preprocessing plays a crucial role. As can be seen, the data was not distributed uniformly. In this manner, we have to pre-process the data with normalization techniques. Normalization makes the optimization problem more numerically stable and makes training less sensitive to the scale of features, so we can better solve for coefficients. When applying normalization, all the values are all now between 0 and 1, and the outliers are eliminate but remain visible within our normalized data. There are two normalization techniques that can be utilized and each of them has its own consequences, but for now, any of them is sufficient.

Mean Normalization Formula

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Min-Max Normalization Formula

$$X_{new} = \frac{X - X_{mean}}{X_{max} - X_{min}}$$

3.2 Normalized Data

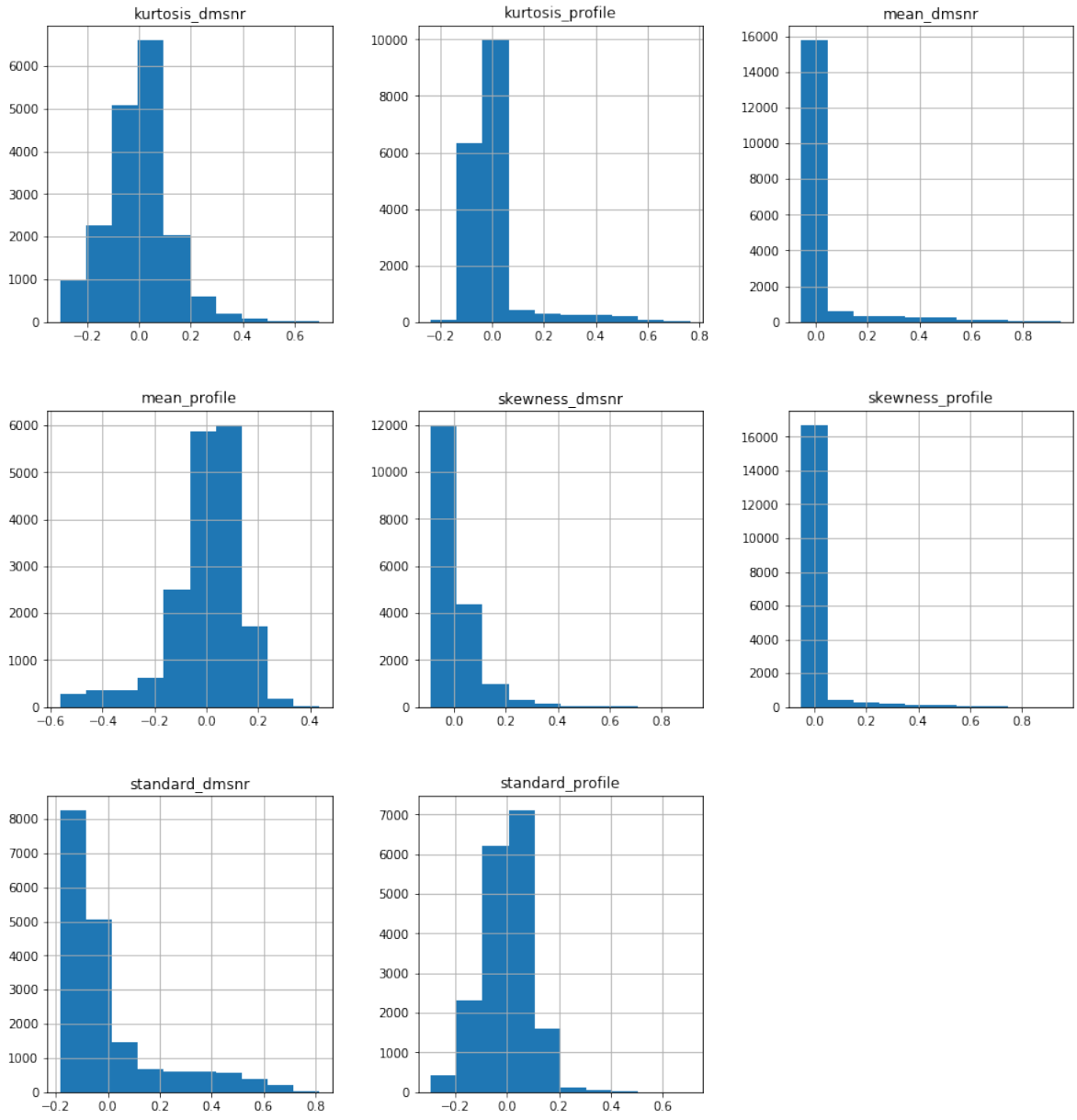

Figure 4: Input Data After Normalization (Min-Max)

3.3 Data Splitting (Working in Progress)

The Idea: the data was randomly shuffled and then the dataset was split into training, validation, and test. Default to 70% of the dataset was allocated for training, 20% was allocated for validation and 10% for testing. The splitting content may change later on.

4 Data Analysis

4.1 To be continue...

5 Modelling

5.1 To be continue...

6 Evaluation

6.1 To be continue...

7 Conclusion

7.1 To be continue...

References

[1] Calla Cofield. *What Are Pulsars?* Apr. 2016. URL: https://www.space.com/32661-pulsars.html. (accessed: 2.20.2020).

[2] Dr Robert Lyon. *HTRU2 Data Set*. Feb. 2017. URL: https://archive.ics.uci.edu/ml/datasets/HTRU2. (accessed: 2.20.2020).

[3] Pavan Raj. *Predicting a Pulsar Star*. May 2018. URL: https://www.kaggle.com/pavanraj159/predicting-a-pulsar-star. (accessed: 2.20.2020).