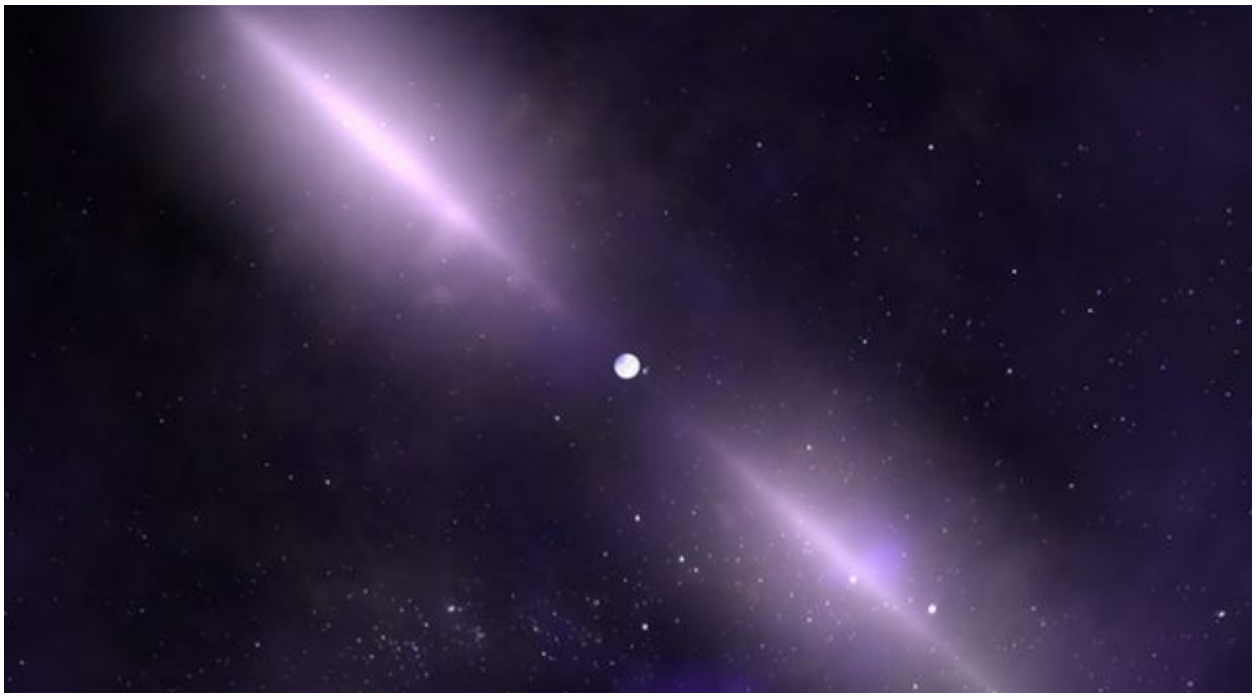
PREDICTING A PULSAR STAR

Introduction:



HTRU2 is a data set that describes a sample of pulsar candidates collected during the High Time Resolution Universe Survey.

Pulsars are a rare type of Neutron star that produces radio emission detectable here on Earth. They are of considerable scientific interest as probes of space-time, the interstellar medium, and states of matter.

As pulsars rotate, their emission beam sweeps across the sky, and when this crosses our line of sight produces a detectable pattern of broadband radio emission. As pulsars rotate rapidly, this pattern repeats periodically. Thus pulsar search involves looking for periodic radio signals with large radio telescopes.

Each pulsar produces a slightly different emission pattern, which varies slightly with each rotation. Thus a potential signal detection known as a 'candidate', is averaged over many rotations of the pulsar, as determined by the length of observation. In the absence of additional info, each candidate could potentially describe a real pulsar. However, in

practice, almost all detections are caused by radio frequency interference (RFI) and noise, making legitimate signals hard to find.

Machine learning tools are now being used to automatically label pulsar candidates to facilitate rapid analysis. Classification systems, in particular, are being widely adopted, which treat the candidate data sets as binary classification problems. Here the legitimate pulsar examples are a minority positive class, and spurious examples the majority negative class.

The data set shared here contains 16,259 spurious examples caused by RFI/noise and 1,639 real pulsar examples. These examples have all been checked by human annotators.

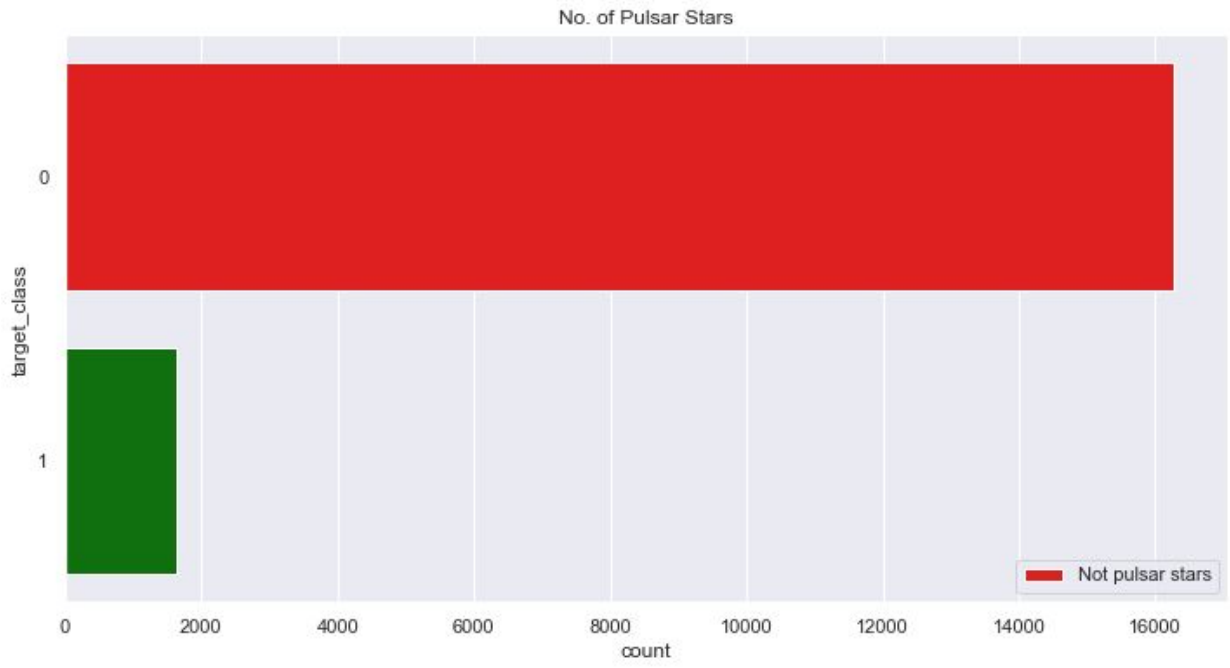
Each row lists the variables first, and the class label is the final entry. The class labels used are 0 (negative) and 1 (positive).

Data Required

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:

Mean of the integrated profile. The standard deviation of the integrated profile. Excess kurtosis of the integrated profile. The skewness of the integrated profile. Mean of the DM-SNR curve. The standard deviation of the DM-SNR curve. Excess kurtosis of the DM-SNR curve. The skewness of the DM-SNR curve. Class HTRU 2 Summary 17,898 total examples. 1,639 positive examples. 16,259 negative examples.

Source: https://archive.ics.uci.edu/ml/datasets/HTRU2



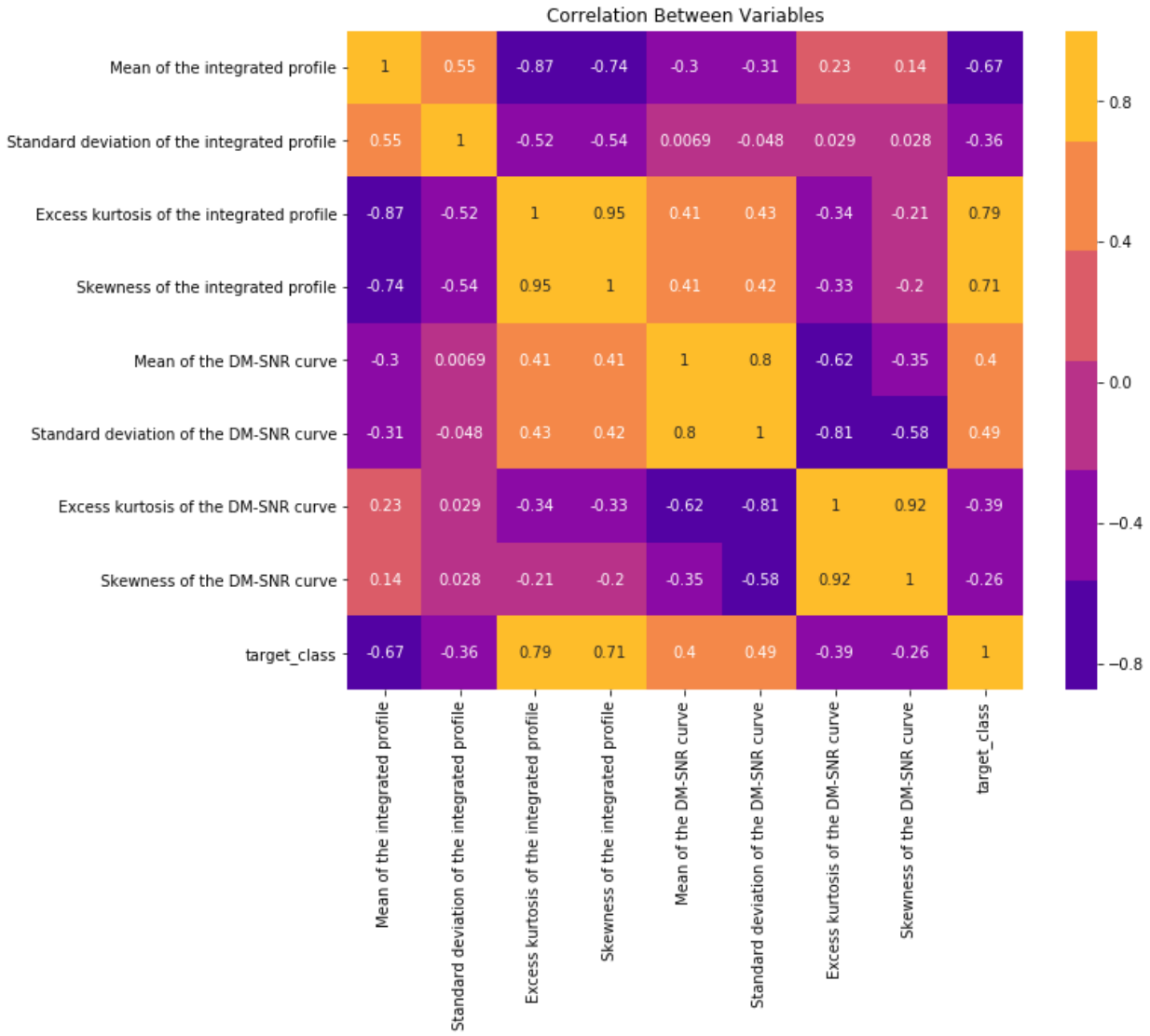
Credit

Data from:: https://archive.ics.uci.edu/ml/datasets/HTRU2

Dr Robert Lyon University of Manchester School of Physics and Astronomy Alan Turing Building Manchester M13 9PL United Kingdom robert.lyon '@' manchester.ac.uk

Methodology

Data analysis and machine learning have become an integrative part of the modern scientific methodology, offering automated procedures for the prediction of a phenomenon based on past observations, unraveling underlying patterns in data and providing insights about the problem. Yet, caution should avoid using machine learning as a black-box tool, but rather consider it as a methodology, with a rational thought process that is entirely dependent on the problem under study. In particular, the use of algorithms should ideally require a reasonable understanding of their mechanisms, properties and limitations, in order to better apprehend and interpret their results. Accordingly, the goal of this thesis is to provide an in-depth analysis of random forests, consistently calling into question each and every part of the algorithm, in order to shed new light on its learning capabilities, inner workings and interpretability. The first part of this work studies the induction of decision trees and the construction of ensembles of randomized trees, motivating their design and purpose whenever possible. Our contributions follow with an original complexity analysis of random forests, showing their good computational performance and scalability, along with an in-depth discussion of their implementation details, as contributed within Scikit-Learn


Correlation Between Variables
Mean of the integrated profile
Standard deviation of the integrated profile
Excess kurtosis of the integrated profile
Skewness of the integrated profile
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
target_class
1 0.55 -0.87 -0.74 -0.3 -0.31 0.23 0.14 -0.67
0.55 1 -0.52 -0.54 0.0069 -0.048 0.029 0.028 -0.36
-0.87 -0.52 1 0.95 0.41 0.43 -0.34 -0.21 0.79
-0.74 -0.54 0.95 1 0.41 0.42 -0.33 -0.2 0.71
-0.3 0.0069 0.41 0.41 1 0.8 -0.62 -0.35 0.4
-0.31 -0.048 0.43 0.42 0.8 1 -0.81 -0.58 0.49
0.23 0.029 -0.34 -0.33 -0.62 -0.81 1 0.92 -0.39
0.14 0.028 -0.21 -0.2 -0.35 -0.58 0.92 1 -0.26
-0.67 -0.36 0.79 0.71 0.4 0.49 -0.39 -0.26 1
Mean of the integrated profile
Standard deviation of the integrated profile
Excess kurtosis of the integrated profile
Skewness of the integrated profile
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
target_class
0.8
0.4
0.0
−0.4
−0.8

Random Forest Classifier

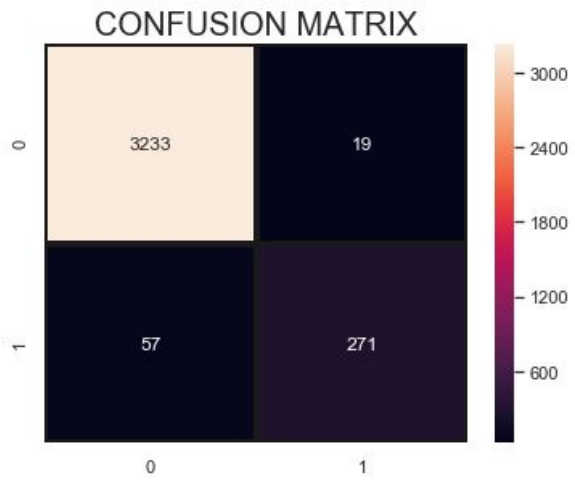
Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object. It is an **ensembled algorithm.** ***Ensembled algorithms*** *are those which combines more than one algorithms of same or different kind for classifying objects.*

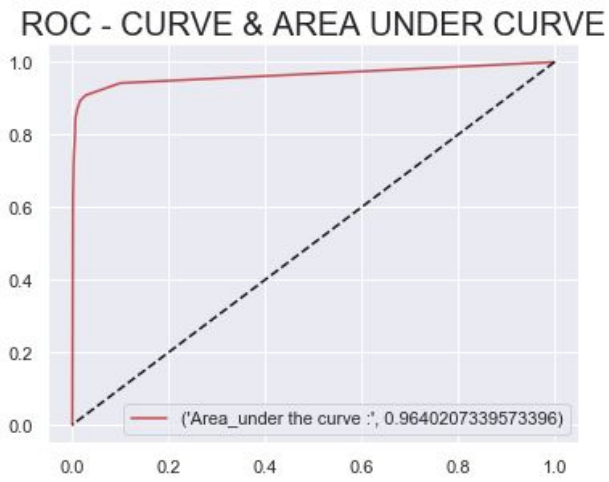
Code for Random Forest Classifier

Here, **n_estimators = Number of decision trees used.**

Evaluating our model

Accuracy for Random Forest Classifier





Conclusion

ere is likely room for further improvement, but this is a big improvement over the best decision tree error of 250,000. There are parameters which allow you to change the performance of the Random Forest much as we changed the maximum depth of the single decision tree. But one of the best features of Random Forest models is that they generally work reasonably even without this tuning.

You'll soon learn the XGBoost model, which provides better performance when tuned well with the right parameters (but which requires some skill to get the right model parameters).