CITY
UNIVERSITY OF LONDON
EST 1894

Pulsar Detection with Naïve Bayes and Random Forests

Stephanie Jury and Anastasios Katsikogiannis

Motivation and Problem Description

Although one of the oldest sciences, big data is a growing challenge in astronomy. It is predicted that, on completion, the Square Kilometre Array ("SKA") of radio telescopes will produce approximately one exabyte (one billion gigabytes) of data per day, equivalent to the entire contents of the internet.[1] Today, many modern telescopes are capable of collecting data faster than humans are able to use it, necessitating the application of machine learning techniques in the domain.[2], [3], [4] One current area of active research is in the detection and discovery of Pulsars, highly magnetised, rotating stars that emit beams of electromagnetic radiation. Although their emission generates detectable patterns across the broadband radio spectrum, a large number of pulsar-like signals are also detected by radio telescopes during a typical observation. This radio frequency interference constitutes noise which masks legitimate signals, making legitimate pulsar discoveries challenging.[5] Given the success of Naïve Bayes (NB) and Random Forest (RF) classifiers in the literature,[6], [7] we apply these two methods to the problem of accurate pulsar detection.

Dataset Description and Preliminary Analysis

The 'High Time Resolution Universe' (HTRU) survey was carried out in 2010 by Keith et al. using the Parkes Observatory 64m radio telescope, with the aim of increasing our understanding of short-duration, millisecond pulsars (MSPs).[8] The 'HTRU2' dataset generated contains 17,898 total examples of pulsar candidates, each described by eight continuous variables. The first four describe the shape of the integrated pulse profile generated by the pulsar candidate, being profile mean, standard deviation, excess kurtosis and skew. The remaining four similarly describe the shape of the de-dispersed signal to noise ratio (DM-SNR) curve associated with the candidate's emission detection, being DM-SNR mean, standard deviation, excess kurtosis and skew. Each candidate has been assigned a binary class by human annotators, labeled either noise (0 or negative) or a real pulsar example (1 or positive). Findings from our preliminary analysis:

- The dataset contains no positional information or other astronomical details.
- The dataset was cleaned before publishing and, as such, contains no missing values.
- All feature variables are continuous and do not require binning.
- Summary statistics for the feature variables are illustrated in Figure 1, illustrating the disparity in their ranges and the considerable skew exhibited by some.
- Histogram traces with 50 bins represent the feature variable distributions by class and are shown in Figure 2. The variables are normalised using the z-score and provide intuition about the shape of the distributions of the features.
- Also evident in Figure 2 is the imbalanced distribution of classes. The dataset contains 16,259 negative examples and 1,639 real pulsar examples, resulting in a majority 'noise' class bias of 90.8%. Such bias has been shown to affect the predictive power of classifiers, as high accuracy can be achieved by simply predicting the majority class.[9] To provide the classifiers exposure to a greater proportion of pulsar classes, we rebalanced the training set by randomly removing noise observations. Creating an equal number of pulsar and noise observations resulted in the removal of 10,212 noise observations, leaving a training set of 2,310 observations in total. The held out test set was not rebalanced to reflect the low instances of pulsar detection in radio telescope observation and likely bias of any new dataset to which the classifier may be applied.
- Variable correlations are shown in Figure 3. Although there is no simple explanation for why certain pulse profile and DM-SNR characteristics should be highly correlated with each other, either positively or negatively, it is clear that pulse profile kurtosis and skew, and DM-SNR mean and standard deviation exhibit high correlation with the class observations. This is considered during our model tuning and feature selection.

Feature	Mean	Standard Deviation	Skew	Minimum	Maximum
Profile mean	111.0800	25.6529	-1.3751	5.8125	192.6172
Profile standard deviation	46.5495	6.8432	0.1266	24.7720	98.7789
Profile excess kurtosis	0.4779	1.0640	3.6381	-1.8760	8.0695
Profile skew	1.7703	6.1679	5.1809	-1.7919	68.1016
DM-SNR mean	12.6144	29.4729	3.6830	0.2132	223.3921
DM-SNR standard deviation	26.3265	19.4706	1.8941	7.3704	110.6422
DM-SNR excess kurtosis	8.3036	4.5061	0.4415	-3.1393	34.5398
DM-SNR skew	104.8577	106.5145	2.7343	-1.9770	1910.0008

Figure 1 – Overview of feature variable continuous distribution parameters.

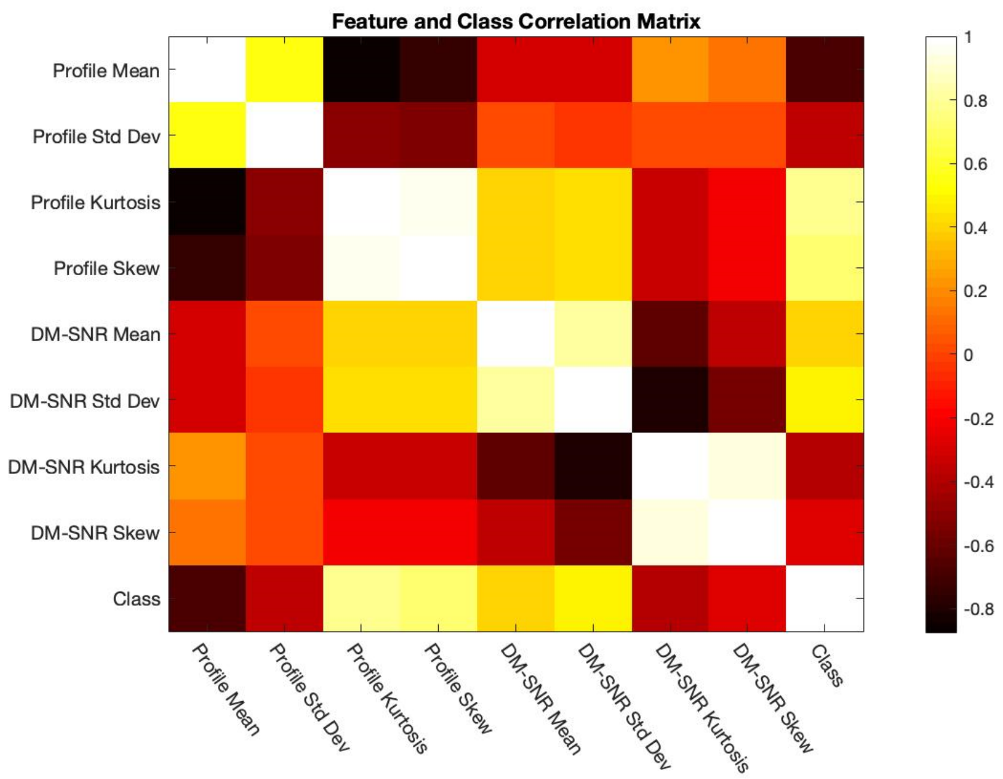

Figure 3 – Correlation plot of features and classes.

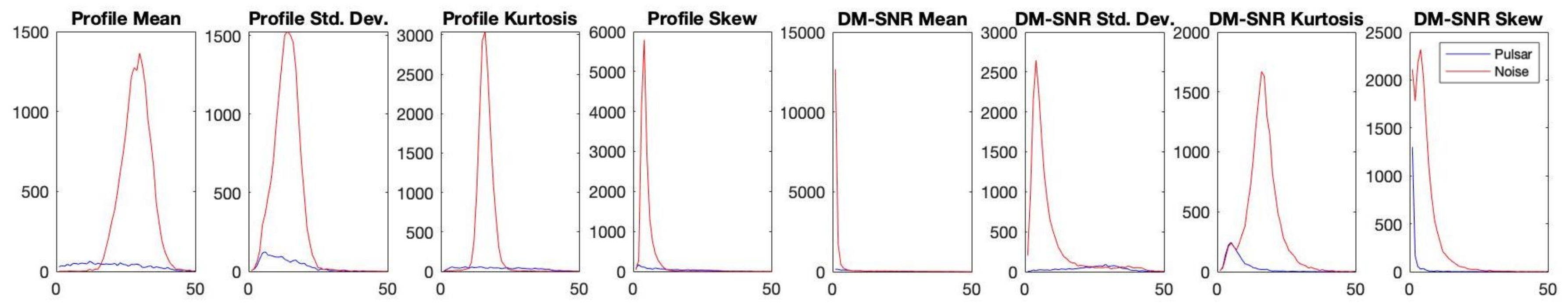

Figure 2 – Normalised feature distributions by class.

Model Overview and Application

Naïve Bayes

Naïve Bayes classifiers ("NB") use Bayes' theorem to predict the conditional probability of an observation's class given the features that describe that observation. The model takes a set of probabilities for each feature given its class ("likelihood") and multiplies these by the class probabilities ("priors") in the dataset. The method is "naïve" as it assumes features are independent of one another given a class ("Conditional Independence Assumption", "CIA"). Likelihoods are often estimated by probability distributions, for example a Gaussian over continuous variables or Multinomial over discrete variables, while priors can be estimated from the data or with domain knowledge. The probability of a new observation belonging to a particular class given its features ("posterior") is used in conjunction with a decision rule to assign the best class prediction. In this investigation we apply the Maximum a Posteriori ("MAP") decision rule which assigns the observation's predicted class according to the largest posterior probability.

Pros

- Performs well even when the CIA is violated.[11]
- Only requires a small number of training data to estimate the parameters necessary for classification and so can be easily trained, even with small datasets.[12]
- Simple to build and fast to train, with theoretical time complexity of O(Np), where N is the number of training examples and p is the number of features.[13]
- Fast at classifying new observations as the model is an "eager learner".
- Not sensitive to irrelevant features.
- Highly scalable and can work well with high-dimensional datasets.[14]

Cons

- Often outperformed by more sophisticated models.[12]
- Posterior probabilities are not mathematically accurate if the CIA is violated.[15]
- If a feature is absent in all observations of a particular class in the training set, it will have a likelihood of zero. Multiplication during inference will result in the model failing however, this can be mitigated by applying techniques such as Laplace smoothing.

Random Forest

Random Forests ("RF") are an ensemble learning method for classification and regression, first implemented by Tin Kam Ho in 1995.[16] The algorithm randomly selects, with replacement, samples of the dataset to generate a large number of new datasets the same size as the original ("bootstrapping") whilst simultaneously recording which observations are omitted in each bootstrapped dataset ("out-of-bag samples"). For each bootstrapped dataset, multiple different decision trees are grown by randomly selecting a subset of the features to act as the root and nodes (typically √n features for a classification problem with n features). The RF is tested on the out-of-bag-samples and the class of each unseen observation determined by the mode class output by all of the individual trees. In a regression task, the output is the mean of all the outputs. The combination of bootstrapping and aggregating the decision is known as "bagging".

Pros

- Performs well even without hyper-parameter tuning.
- Can handle large numbers of features even in the presence of complex interactions.[19]
- Corrects for single decision trees' high variance and overfitting as multiple trees are decorrelated.
- Fast at classifying new observations as, like NB, the model is an "eager learner".
- Inherits decision trees' robustness to outliers as atypical observations are isolated into small leaves.[18]

Cons

- Often outperformed by more sophisticated models.[12]
- Can be slow to train as time complexity scales with the number of grown trees.
- If the number of relevant features is considerably lower than the total number of features (n), subsampling will result in multiple trees being grown with low predictive accuracy.

We split the data into a training set (70%) and test set (30%) then rebalanced the training set by removing 10,212 observations of the noise class; the test set was left imbalanced. We investigated multiple model parameters and tested their effect on mean classification error using 10-fold cross validation. The parameters applied to the held out test data were those that resulted in the lowest mean classification error over the ten runs.

Choice of Parameters and Experimental Results

Naïve Bayes

In tuning the NB model we investigated the effect of changing three different model parameters. Their impact on the mean classification error rate over ten runs in validation is illustrated in Figure 3.

- Changing the prior from a uniform relative frequency distribution over the two classes, to progressively more biased relative frequency distributions representing the relative scarcity of pulsar observations in the dataset.
- Using different combinations of Gaussian and kernel density estimation ("KDE") for the class-conditional probabilities of the features. Starting with all features assumed Gaussian, KDE was progressively applied to features based on their skew.
- Changing the kernel smoother window width to those models using KDE where the mean classification error rate was already less than 0.0950 (9.5%).

Results

- Bias in the prior in excess of 60% noise to 40% pulsars had a negative impact on accuracy in validation.
- Applying KDE to features improved accuracy in validation. The lowest mean error rates were achieved, across almost all instances, when KDE was applied to those features exhibiting skew greater than or equal one and a Gaussian applied to the remainder.
- Setting kernel smoother width at one or higher had considerable negative impact on accuracy in validation. The best mean accuracy was achieved with kernel smoother width of 0.1.
- The lowest mean error rate achieved during ten-fold cross validation was 0.0764, corresponding to an accurate classification rate of 92.06%.
- Averaged over 10 validation sets, all models took less than 0.1 seconds to build.

Random Forest

In tuning the RF model we investigated the effect of changing two different model parameters. Their impact on the mean classification error rate over ten runs in validation is illustrated in Figure 4.

- Changing the number of trees grown in the ensemble.
- Changing the minimum number of observations per tree leaf (the "minimum leaf size").

Results

- Figure 5 demonstrates that the mean classification error during validation decreased markedly as the number of trees grown increased from one to 20 but from 20 onwards there was little incremental improvement.
- The surface plot in Figure 4 demonstrates that the minimum leaf size and number of grown trees are not independent in their contribution to mean classification error. The slope of the surface indicates that when the number of trees grown was greater than approximately 10, the mean classification error decreased as minimum leaf size decreases.
- The lowest mean error rate occurred at the minimum indicated on the surface plot. This demonstrates that the optimal RF model during validation was generated with 80 grown trees and minimum leaf size of one.
- The lowest mean error rate achieved during ten-fold cross validation was 0.0550, corresponding to an accurate classification rate of 94.50%.
- The RF runtime was directly proportional do the number of trees grown, as shown in Figure 6, with *runtime* ≃ 0.0268 × (*number of trees grown*). The runtime increased slightly as the minimum leaf size was reduced, but this effect was overpowered by the number of grown trees.
- Averaged over 10 validation sets, the optimal RF model took 2.63 seconds to build.

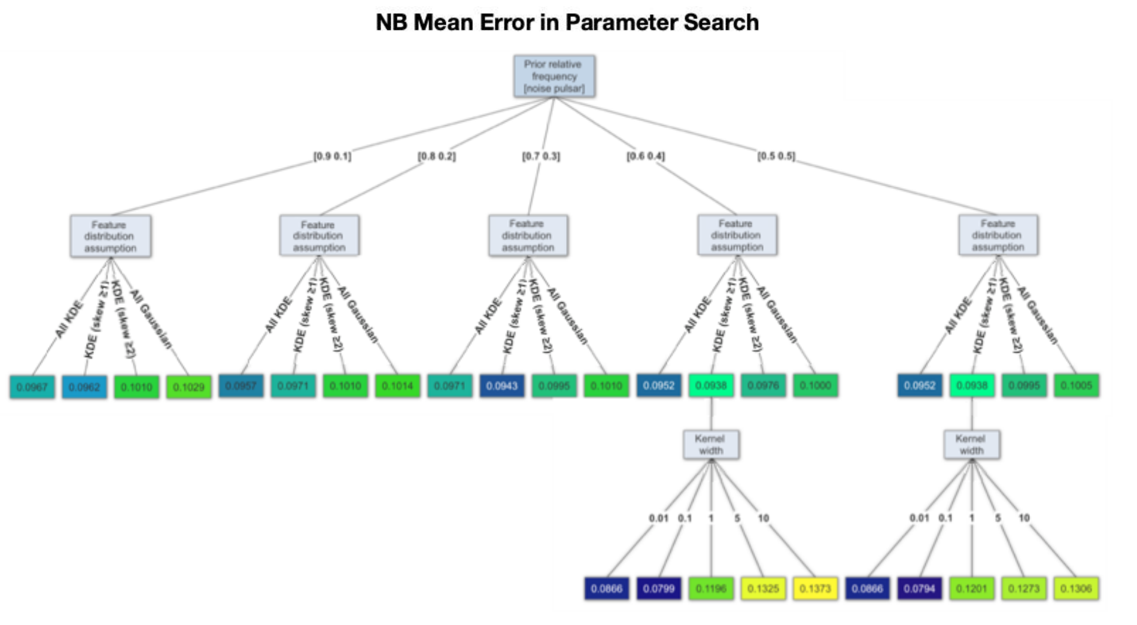

Figure 3 – Naïve Bayes model hyperparameter search and resulting mean classification error rates over ten-fold cross validation.

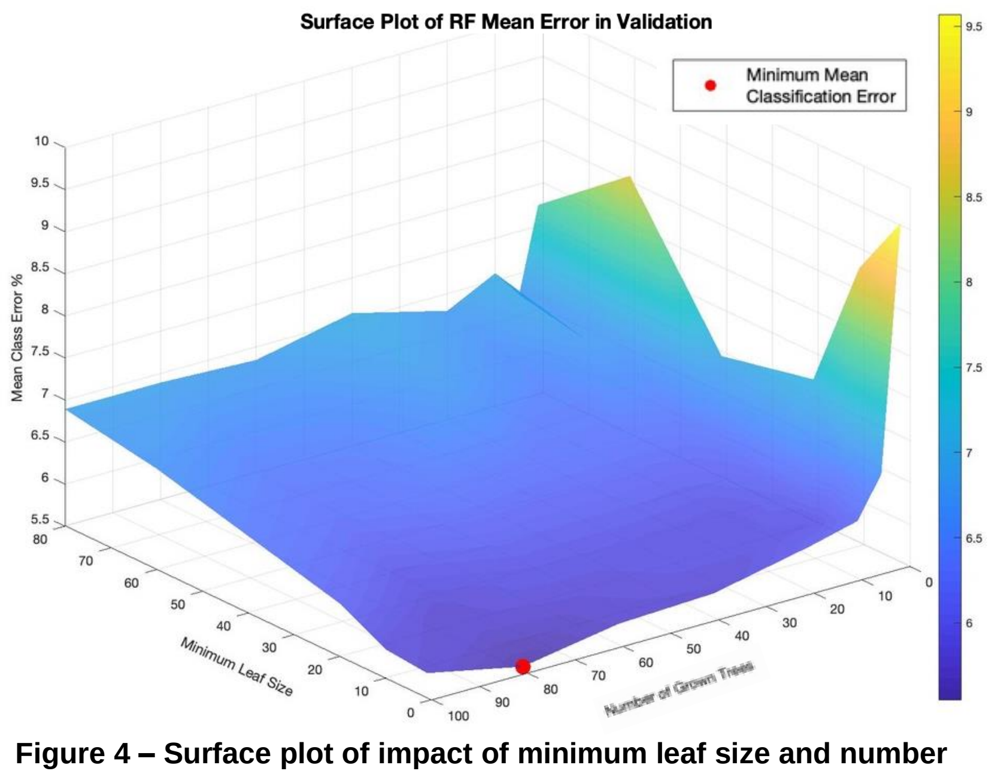

Figure 4 – Surface plot of impact of minimum leaf size and number of grown trees in the random forest ensemble, on the mean classification error percentage during ten-fold cross validation.

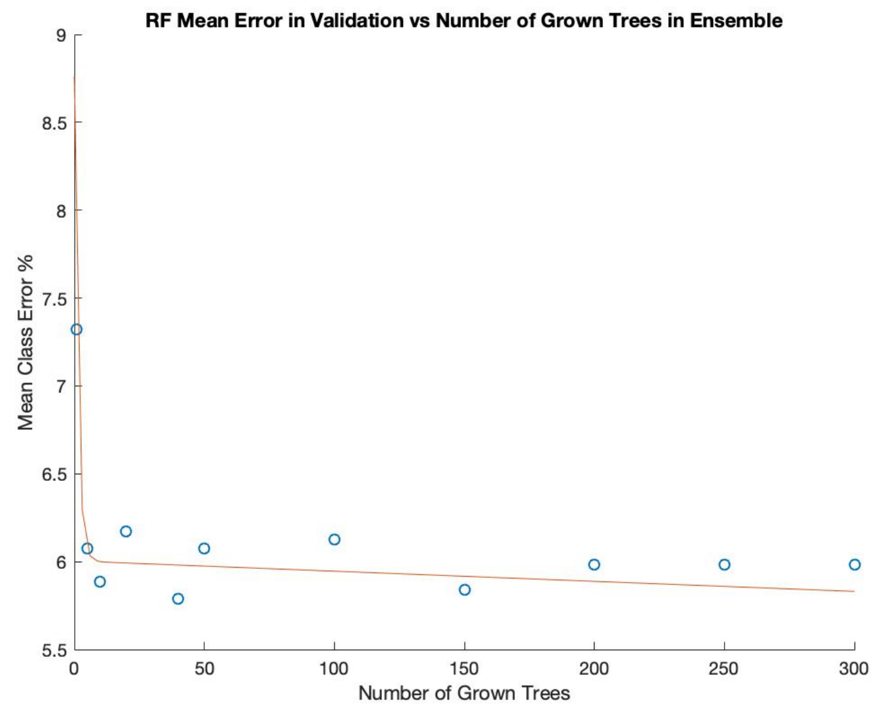

Figure 5 – Impact of number of grown trees in the random forest ensemble on mean classification error percentage in ten-fold cross validation.

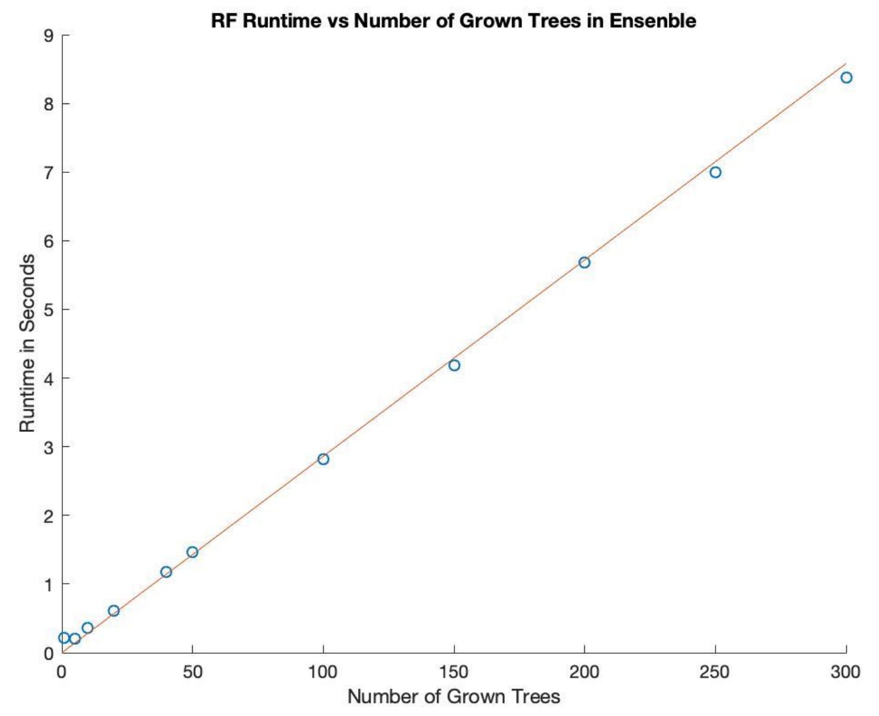

Figure 6 – Impact of number of grown trees in the random forest ensemble on model runtime.

Hypothesis

Work by Bates et al., achieved pulsar detection accuracy of 85% by applying sophisticated artificial neural networks to the 2.5 million unfiltered observations generated by the HTRU survey.[22] Although "out of the box" RF and NB models are unlikely to achieve similar under the same test conditions, the literature demonstrates they can perform very well, especially when applied to clean or pre-processed datasets such as HTRU2.[21] It is generally the case in the literature that calibrated RF models achieve greater classification accuracy than NB models.[12], [20] In fact, RF models are shown empirically to perform among the best performing classifiers.[20] It is our hypothesis that a RF model will generate a lower classification error percentage than NB when applied to the HTRU2 dataset. It is important to note however, that model performance does vary considerably across datasets,[21] and that the class bias we observed may impact the performance of each model differently.[5], [22]

Analysis and Evaluation of Results

Consistent with the literature and supporting our hypothesis, our experiments demonstrated that a tuned RF model outperforms a tuned NB model, achieving a lower classification error when applied to the HTRU2 dataset. This was demonstrated by RF versus NB classification accuracy rate of 94.3% to 92.1% in validation and 96.4% to 95.8% in testing, as illustrated in in Figure 7 and is corroborated by the receiver operating characteristic ("ROC") curves in Figure 8, with areas under the curve of 98.0% and 97.2% for RF and NB respectively.

We were surprised to see that both models achieved higher classification accuracy in testing, as it was the case in the literature that models applied to this dataset performed better in validation.[22] We believe this is a consequence of how we managed dataset bias during model training. In Figure 7 it is clear that both models in validation achieved superior positive predictive values ("PPVs") (NB: 95.63%, RF: 97.43%) to negative predictive values ("NPVs") (NB: 71.13%, RF: 74.59%), however in testing this phenomenon was reversed, with PPVs (NB: 89.08%, RF: 91.92%) being significantly higher than NPVs (NB: 99.03%, RF: 99.22%). Because the models were trained on unbiased data, when they encounter the considerable negative class bias in the test set, they are prone to predicting a greater proportion of false positive and a lower proportion of false negative instances. This stronger NPV is rewarded by the test set negative class bias, resulting in greater accuracy but the weaker PPV is penalised by the F1 score, being the harmonic mean of the PPV and true positive rate ("TPR"). The relative outperformance in classification accuracy by the RF model can be mostly attributed to its superior ability to assign new observations to the minority class and classify true positive instances.

The dataset parameters were not normally distributed and contained a significant number of outlier values. We investigated the impact of z-score normalisation on both the NB and RF model and determined this made negligible impact on classification accuracy in validation. Before tuning the NB model, we initially tested it using the default parameter settings of a uniform prior distribution over the classes and all parameters approximated using Gaussian distributions. This achieved a classification error rate of 10.05%, indicating that, consistent with the literature,[11] NB is a powerful tool despite its simplicity. Tuning the hyperparameters had only a small impact on the classification error in validation, achieving a further reduction of 2.11%. Training and testing the NB model was computationally very efficient, with each model application in validation taking less than a second, and final application of the tuned model to the test set taking 1.58 seconds. This makes NB easy to apply to large datasets however, there is evidence to suggest that model accuracy does not scale well.[23]

Training the RF model was computationally much more intensive, with the time taken to run each model in validation increasing linearly with the number of trees grown in the ensemble, as shown in Figure 6, whilst the improvement in mean classification error was only seen over the first 80 grown trees, as shown Figure 5. This indicates that RF models are prone to overfitting as the number of trees increases, and that a smaller ensemble size generates a higher classification accuracy whilst being computationally more efficient. Application of the final tuned model to the test set was not meaningfully slower than NB, at 1.93 seconds.

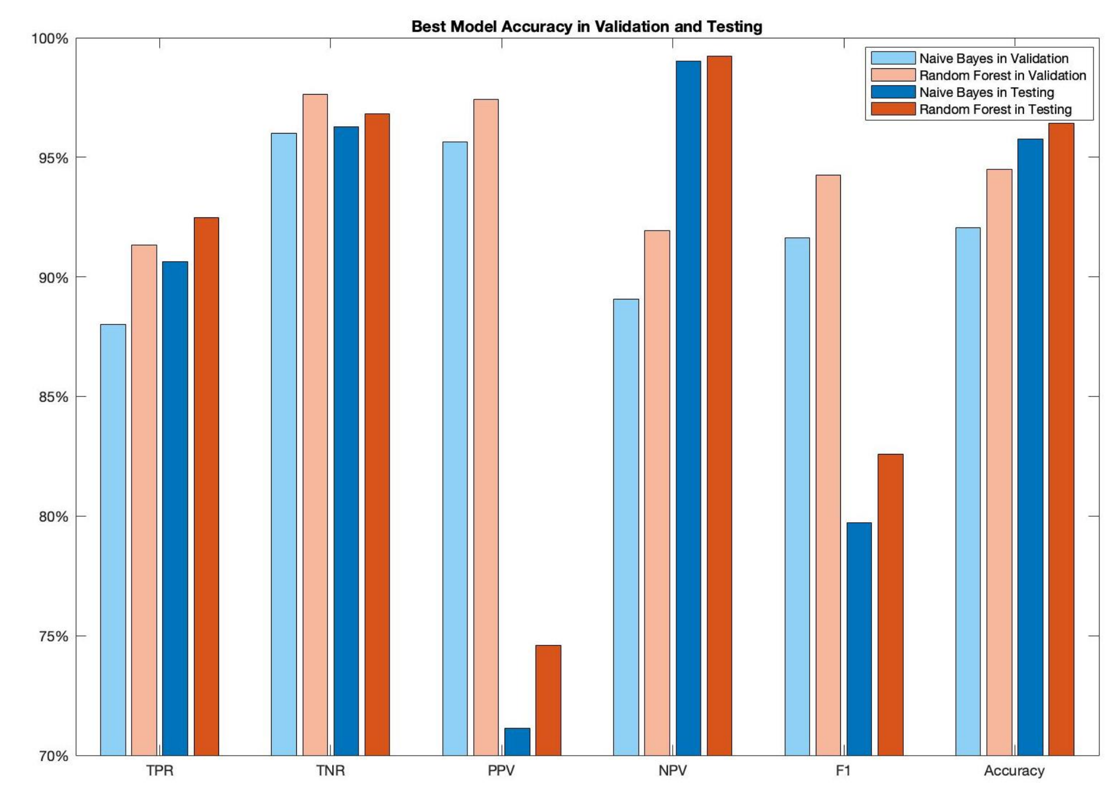

Figure 7 – Error metrics generated by the best models applied in validation and testing.

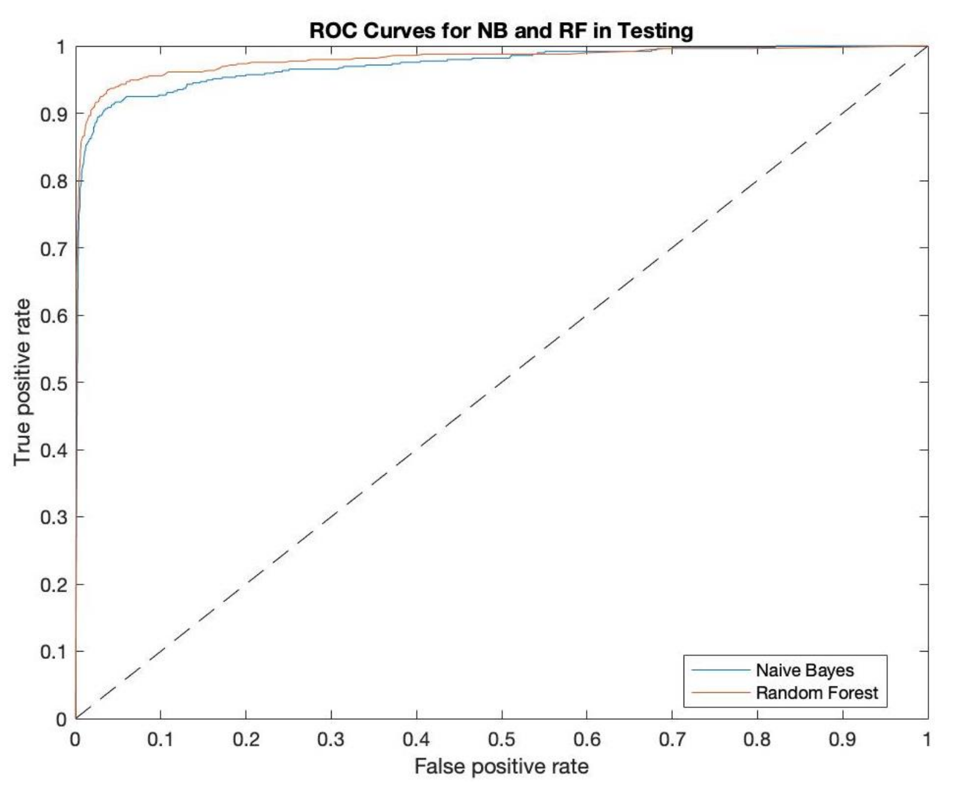

Figure 8 – Error metrics generated by the best models applied in testing.

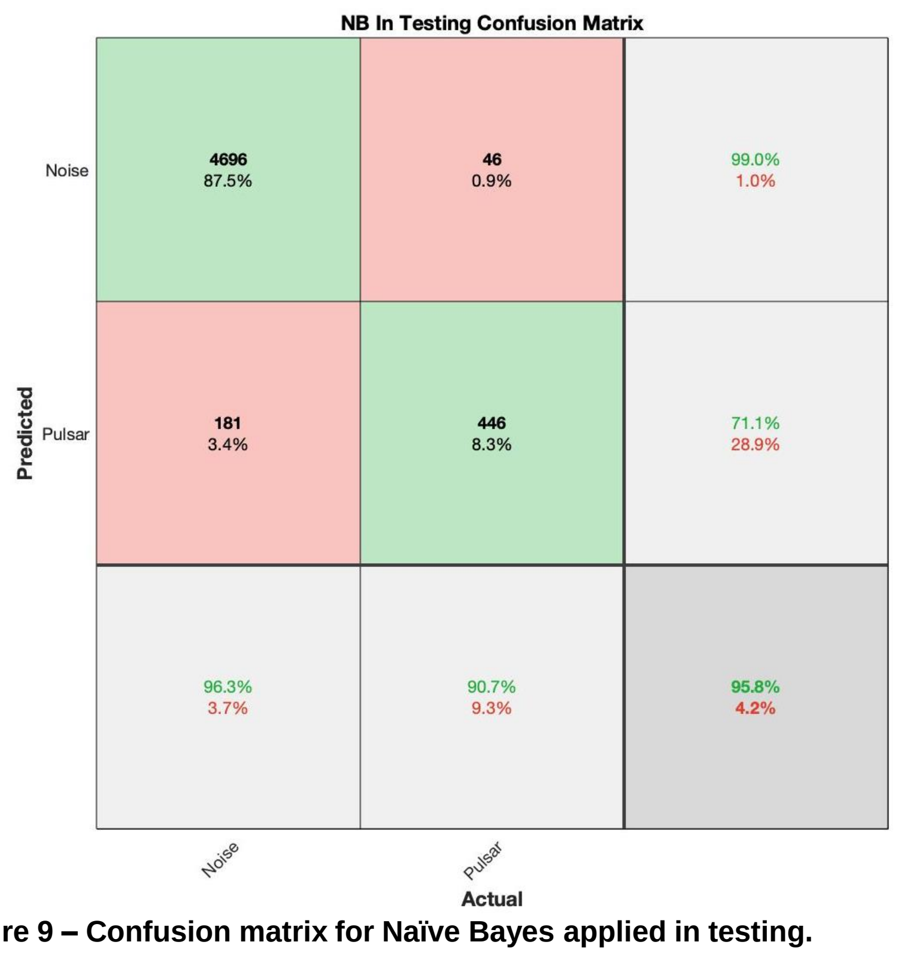

Figure 9 – Confusion matrix for Naïve Bayes applied in testing.

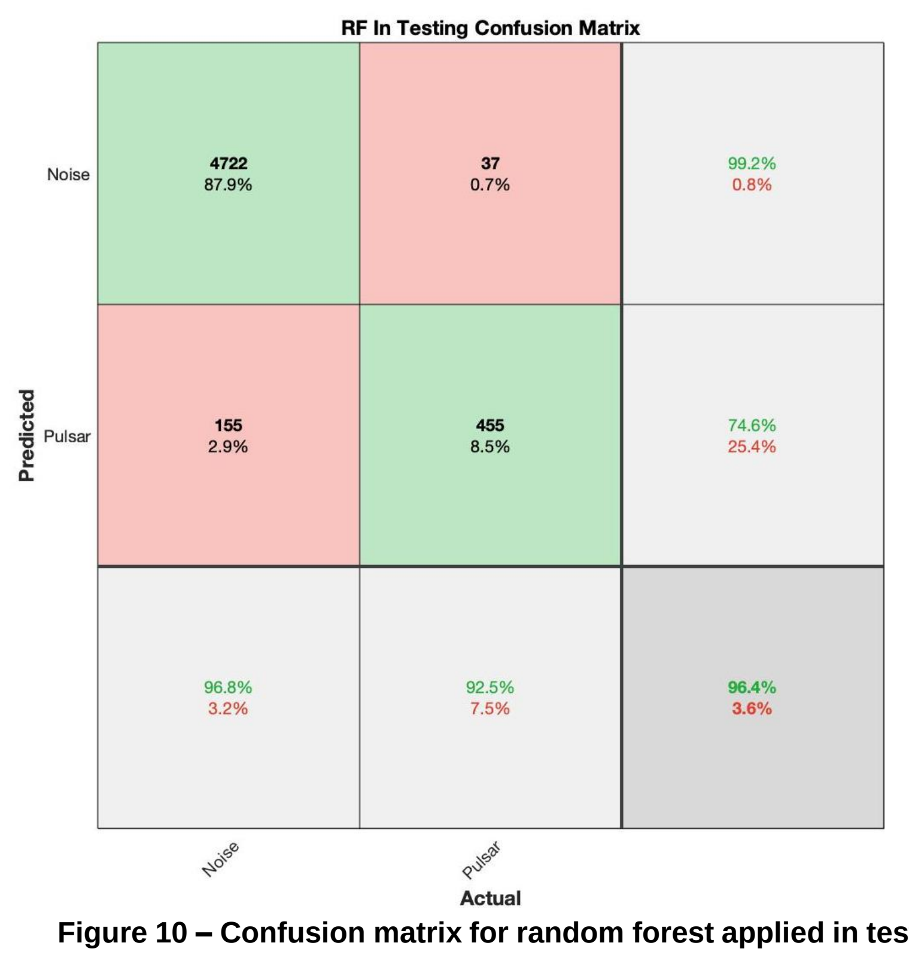

Figure 10 – Confusion matrix for random forest applied in testing.

Lessons Learned and Further Study

Both NP and RF performed well in classifying pulsars in an imbalanced dataset and little hyperparameter tuning or data preprocessing was required, despite the high skew and number of outliers in the parameter data. Setting KDE distribution assumptions for the NB model and setting the appropriate number of grown trees for the RF model achieved good classification accuracy. However, further work should include testing the impact of feature-specific normalisation and outlier handling methods. The negative majority class bias in our dataset resulted in relatively poor positive predictive performance of both models during testing. Although correct identification of noise data is valuable for radio data cleaning, there is greater scientific value in reducing the human cost of identifying new pulsars. In further work we would use a synthetic data generation method, such as SMOTE or ADASYN,[24] to correct the class bias as opposed to random undersampling and test if this improved PPV. Due to the relatively small number of features available in the HTRU2 dataset we did not investigate the impact of feature reduction methods in applying the NB and RF models. Further work might include the analysis of feature importance and the impact of feature reduction on classification accuracy, such that more computationally intensive extensions to these models could be applied to a reduced dataset.

[1] Barwick, H. "Ska telescope to generate more data than entire internet in 2020." *Computerworld Australia* (2011).
[2] Richards, Gordon T., et al. "Efficient photometric selection of quasars from the Sloan Digital Sky Survey: 100,000 z< 3 quasars from Data Release One." *The Astrophysical Journal Supplement Series* 155.2 (2004): 257.
[3] Salzberg, Steven, et al. "Decision trees for automated identification of cosmic-ray hits in Hubble Space Telescope images." *Publications of the Astronomical Society of the Pacific* 107.709 (1995): 279.
[4] Fiorentin, P. Re, et al. "Estimation of stellar atmospheric parameters from SDSS/SEGUE spectra." *Astronomy & Astrophysics* 467.3 (2007): 1373-1387.
[5] Eatough, R. P, et al. "Selection of radio pulsar candidates using artificial neural networks." *Monthly Notices of the Royal Astronomical Society* 407.4 (2010): 2443-2450.
[6] Broos, Patrick S., et al. "A naive Bayes source classifier for X-ray sources." *The Astrophysical Journal Supplement Series* 194.1 (2011): 4.
[7] Zhao, Yongheng, and Yanxia Zhang. "Comparison of decision tree methods for finding active objects." *Advances in Space Research* 41.12 (2008): 1955-1959.
[8] Keith, M. J., et al. "The high time resolution universe pulsar survey–i. system configuration and initial discoveries." *Monthly Notices of the Royal Astronomical Society* 409.2 (2010): 619-627.
[9] Chen, Chao, Andy Liaw, and Leo Breiman. "Using random forest to learn imbalanced data." *University of California, Berkeley* 110 (2004): 1-12.
[10] O'Brien, Tim. "4. Observations of Pulsars." *Properties of Stars*, Jodrell Bank Centre for Astrophysics, 2010, www.jb.man.ac.uk/distance/frontiers/pulsars/section4.html
[11] Hand, David J., and Keming Yu. "Idiot's Bayes—not so stupid after all?." International statistical review 69.3 (2001): 385-398.
[12] Caruana, Rich, and Alexandru Niculescu-Mizil. "An empirical comparison of supervised learning algorithms." Proceedings of the 23rd international conference on Machine learning. ACM, (2006).
[13] Fleizach, Chris, and Satoru Fukushima. "A naive Bayes classifier on 1998 KDD Cup." (1998).
[14] Karim, Md Rezaul, and Sridhar Alla. Scala and Spark for Big Data Analytics: Explore the concepts of functional programming, data streaming, and machine learning. Packt Publishing Ltd., (2017).
[15] Lewis, David D. "Naive (Bayes) at forty: The independence assumption in information retrieval." European conference on machine learning. Springer, Berlin, Heidelberg, (1998).
[16] Ho, Tin Kam. "Random decision forests." Document analysis and recognition, 1995, proceedings of the third international conference on. Vol. 1. IEEE, (1995).
[17] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. The elements of statistical learning. Vol. 1. No. 10. New York, NY, USA:: Springer series in statistics, (2001).
[18] Loureiro, Antonio, Luis Torgo, and Carlos Soares. "Outlier detection using clustering methods: a data cleaning application." Proceedings of KDNet Symposium on Knowledge-based systems for the Public Sector. Springer, (2004).
[19] Strobl, Carolin, James Malley, and Gerhard Tutz. "An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests." Psychological methods 14.4 (2009): 323.
[20] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
[21] Demšar, Janez. "Statistical comparisons of classifiers over multiple data sets." Journal of Machine learning research 7 Jan (2006): 1-30.
[22] Bates, S. D., et al. "The High Time Resolution Universe Pulsar Survey—VI. An artificial neural network and timing of 75 pulsars." Monthly Notices of the Royal Astronomical Society 427.2 (2012): 1052-1065.
[23] Kohavi, Ron. "Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid." *KDD*. Vol. 96. (1996).
[24] Chawla, Nitesh V. "Data mining for imbalanced datasets: An overview." *Data mining and knowledge discovery handbook*. Springer, Boston, MA, (2009). 875-886.