A Comparison of Naïve Bayes and Random Forests Applied to the Cardiotocography Dataset

Lesley Dwyer and Andrea Silas

Brief description and motivation of the problem

- Cardiotocography (CTG) measures the heart rate of the foetus and uterine contractions of the mother and is monitored during pregnancy to check the state of the foetus [1]. The models will predict normal, suspect and pathologic classes of foetal health using CTG data.
- We will compare Naïve Bayes and Random Forest models for a multi-class classification problem to determine which performs better.

Initial analysis of the dataset including basic statistics

- Dataset: Cardiotocography Data Set from UCI Machine Learning Repository
- The data contains 2126 observations. We have split this into 1488 for training and 637 for testing.
- [1] and [2] used 21 variables from the dataset and the 3-class NSP column as the classification. However, [2] only classified the data based on 2 classes and left out the 'suspect' class (labelled as 2) as they felt it did not help the clinicians make a decision. We will use all 3 classes for our model prediction and the 21 variables as features to allow us to compare to the papers referenced.
- Features are: 8 continuous and 13 discrete [2].
- Mean and standard deviation were calculated for all variables for each class. One variable, DS (# severe decelerations per second) was mostly zeroes. However, we have left this in as did our reference papers.
- Skewness was calculated for all variables, and we found that all variables had some level of skew. The variables with the largest skew were FM (# of foetal movements per second), DS (# of severe decelerations per second), DP (# of prolonged decelerations per second) and Nzeros (# of histogram zero).

Brief summary of the two ML models with their pros and cons

Naïve Bayes:

The Naïve Bayes Classifier is a simple probabilistic model combining Bayes theorem and the assumption of independence between attributes with a decision rule, the maximum a posteriori rule. The product of the probabilities of each attribute for each class are worked out and the data sample is then assigned to the class with the highest probability [7].

Pros:

- Known to be very effective and comparable with more sophisticated models, such as decision trees and even neural networks [8] [6].
- Easily understood and implemented.
- The assumption of independence means that we can train on smaller datasets than would otherwise be required [8] [9].
- Not sensitive to irrelevant features [9].
- Fast

Cons:

- Independence assumption can often lead to poor probabilities (yet the classification performance can still be good) [9].
- Prediction is generally less accurate than other models, such as Random Forests [1].

Random Forests

Random Forest is an ensemble algorithm which builds many decision trees using bootstrap aggregation or 'bagging'. Its main difference to bagging is that it only selects from a random sample of features each time it splits a tree, instead of using all features. For classification problems, it takes the majority vote of the decision trees to determine the class of each observation [3].

Pros:

- Improved accuracy over other classification methods [4].
- By only choosing a random subset of features at each tree split, it decorrelates the trees thus improving performance [3].
- Reduced variance when compared to a single decision tree [3].
- They do not overfit as more trees are added [4].

Cons:

- Reduced interpretability when compared to a single tree [3].

Hypothesis statement

- In [1], they compared 7 algorithms, including Naïve Bayes and Random Forests. Random Forest outperformed Naïve Bayes with 93.2% accuracy. Naïve Bayes had 82.31% accuracy.
- Based on the results from [1], we expect Random Forest to outperform Naïve Bayes in our comparison.

Description of choice of training and evaluation methodology

- Both [5] and [6] split the data as 70% Training and 30% Testing, so we have done the same.
- [1] and [2] used 10-fold cross-validation on all algorithms to test performance.
- We used 10-fold cross-validation for Naïve Bayes and out-of-bag error for Random Forest to estimate generalisation error.
- We also performed hyperparameter tuning for each model.
- We evaluated both models by comparing Accuracy, Precision, Recall (Sensitivity) and F-Measure, similar to the reference papers.

Choice of parameters and experimental results

Naïve Bayes parameters: A 'best model' was built choosing the best combination of three hyperparameters; distribution, prior and kernel smoothing window width.
Main Experimental Results: The lowest training error can be attributed to the model with a kernel distribution of width of approximately 0.2 and a prior dictated by the relative frequencies of each class (see figure 2). This produced a training error of 6.04% and a test error of 7.23%. In this case, precision and recall for class 1 (normal) are 98.78% and 93.64%, respectively. The recall of class 3 (pathologic) is 92.5% and precision is 67.27%, indicating that many true pathologic values are being predicted, but also too many are being classified as pathologic.

Random Forests parameters: A grid search was used to tune two hyperparameters: number of predictors chosen and number of trees built.
Main Experimental Results: The hyperparameters with the lowest out-of-bag error were: number of predictors = 5 and number of trees = 250. This produced a training error of 5.64% and test error of 5.65%. The values for precision, recall and F-measure were fairly consistent across the three classes.

Test performance comparison

Random Forest Accuracy = 94.35 % Naïve Bayes Accuracy = 92.77 %

Model	Class	Precision	Recall	F-measure
Random Forest	Normal	96.00%	97.56%	96.77%
Naïve Bayes	Normal	98.78%	93.64%	96.14%
Random Forest	Suspect	84.71%	80.00%	82.29%
Naïve Bayes	Suspect	75.56%	87.18%	80.96%
Random Forest	Pathologic	94.23%	89.09%	91.59%
Naïve Bayes	Pathologic	67.27%	92.50%	77.89%

Analysis and critical evaluation of results

- As expected, Random Forest outperformed Naïve Bayes in terms of accuracy, but not by as much as in [1].
- Our Naïve Bayes performed better on precision for the normal class and recall for the suspect and pathologic classes than Random Forest. However, in general Random Forest was more consistent across the three measures.
- Our Random Forest model accuracy of 94.35% was comparable to [1] which was 93.20%. [3] indicates that a good choice for the number of predictors in a Random Forest model is the square root of the total number of features. The square root of 21 features is 4.6, and the optimal number for our model did come out to 5.
- Our Naïve Bayes model performed better on all measures than theirs in [1] did. This could be due to hyperparameter optimisation, which was not mentioned in the papers, or a different software package being used.
- In our Random Forest model, the classification error went down (see figure) as the number of trees increased, which was expected [3].
- As in [1] and [7] our Naïve Bayes classifier predicted more accurately on the normal class than on the suspicious or pathologic.
- Even though our reference papers used 10-fold cross validation, out-of-bag error was listed as a valid alternative to calculate generalisation error for Random Forests without the bias present in cross-validation [4]. Indeed, out-of-bag error produced during hyperparameter optimisation was very similar to the training error and was a good prediction of the test accuracy for our model.
- Speed is negligible in the comparison of these models, however, if additional hyperparameters were evaluated, this could change.

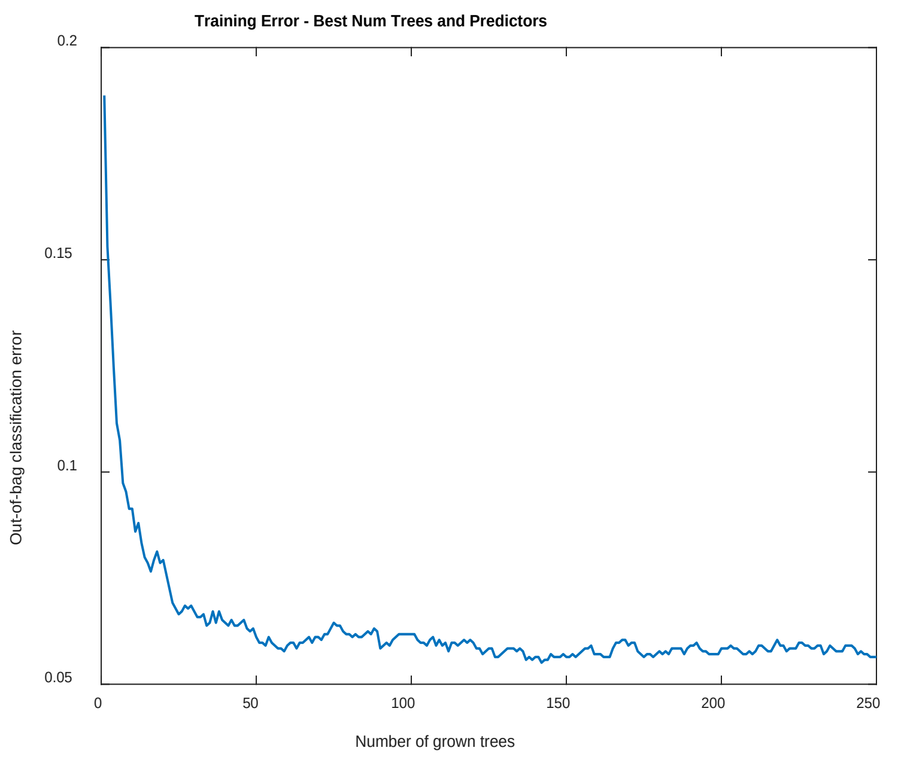

Figure 1. Training Error for optimal hyperparameters

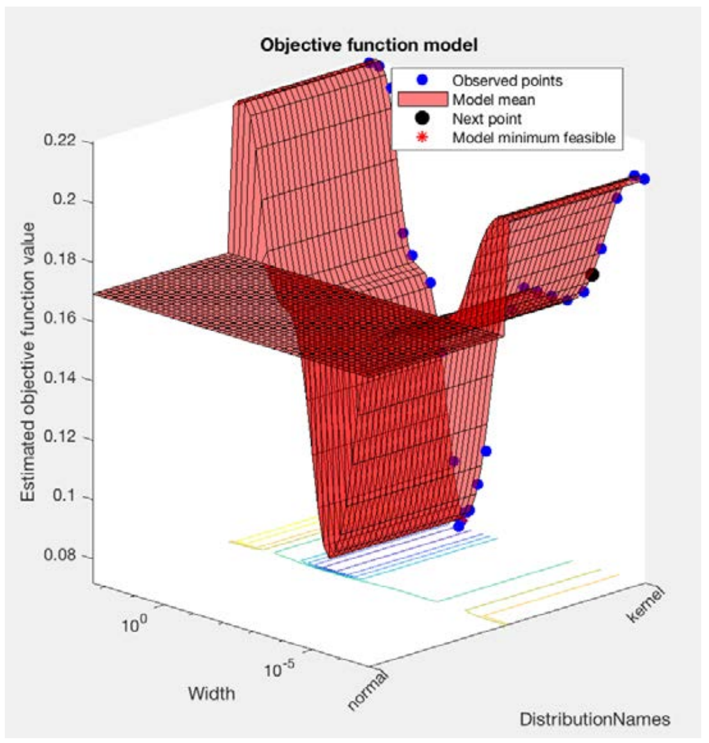

Figure 2. Optimised version of Naïve Bayes Model

Lessons learned and future work

- Feature selection may further improve performance on Naïve Bayes as shown in [6]. Although this also was attempted in [5] for Random Forests, accuracy was not shown to improve when the number of features was reduced.
- There are further hyperparameters that can be tuned for Random Forests, such as optimal leaf size, as mentioned in [5] which may result in improved accuracy.

References

[1] D. Bhatnagar and P. Maheshwari, 'Classification of Cardiotocography Data with WEKA', *International Journal of Computer Science and Network*, vol. 5, no. 2, pp. 412-418, 2016. [Online]. Available at: http://eprints.rclis.org/29886 (Accessed on: 29 October 2018)
[2] H. Sahin and A. Subasi, 'Classification of the cardiotocogram data for anticipation of fetal risks using machine learning techniques', *Applied Soft Computing,* vol. 33, no. 1568-4946, pp. 231-238, 2015. [Online]. Available at: https://doi.org/10.1016/j.asoc.2015.04.038 (Accessed on: 29 October 2018)
[3] G. James, D. Witten, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning with Applications in R, New York: Springer, pp. 316-321, 2017.
[4] L. Breiman, 'Random Forests', *Machine Learning,* vol. 45, no. 1, p. 5–32, 2001. [Online]. Available at: https://link.springer.com/article/10.1023/A:1010933404324 (Accessed on: 15 November 2018)
[5] M. Arif, 'Classification of cardiotocograms using random forest classifier and selection of important features from cardiotocogram signal', *Biomaterials and Biomedical Engineering,* vol. 2, no. 3, pp. 173-183, 2015. [Online]. DOI: 10.12989/bme.2015.2.3.173 (Accessed on: 29 October 2018)
[6] M. E. B. Menai, F. J. Mohder and F. Al-mutairi, 'Influence of Feature Selection on Naïve Bayes Classifier for Recognizing Patterns in Cardiotocograms', *Journal of Medical and Bioengineering,* vol. 2, no. 1, 2013. [Online]. Available at: http://www.jomb.org/uploadfile/2013/0424/20130424041805327.pdf (Accessed on: 29 October 2018)
[7] C. Sundar, M. Chitradevi, G. Geetharamani, "An Analysis on the Performance of Naïve Bayes Probabilistic Model Based Classifier for Cardiotocogram Data Classification," *International Journal on Computational Sciences & Applications,* vol. 3, no. 1, pp. 17-26, 2013. [Online]. Available at: https://wireilla.com/papers/ijcsa/V3N1/3113ijcsa03.pdf (Accessed on: 29 October 2018)
[8] T. Mitchell, Machine Learning. Singapore: McGraw – Hill Book Co, p.177, 1997.
[9] C. Bishop, Pattern Recognition and Machine Learning, New York: Springer Science + Business Media, p.381. 2009.