Identifying the health of fetal using Cardiotocography (CTG) data and machine learning techniques

Nikhil Kumar Reddy Badveli

Abstract—Child Mortality rate is one of the key indicators of human progress. To improve this metric, access to fetal health information is vital and it is made possible by data collected using Cardiotocographs (CTGs). This case study utilizes such data and predicts one of three possible classes - *Normal, Suspect and Pathological* that depicts the risk associated for a particular child. The data is found out to be highly imbalanced with the *Normal* class dominating at almost 80% of the total number of samples, which is to be expected since the cases with high risk are fewer in general. As such, an oversampling technique called SMOTE is used to boost up the number of samples for the two minority classes. And performance metrics like Precision, Recall, F1-score other than just classification accuracy are used to get a more complete picture. Different classifiers such as SVM, kNN etc., from the *sklearn* package were tested for this problem. Before oversampling, although high accuracy was achieved, all the classifiers performed poorly in the other metrics due to the nature of an imbalanced dataset. A noticeable jump in all the performance metrics is observed when tested after oversampling, with the accuracy reaching upto approx.98%.

Index Terms—CTG, Fetal health, Imbalanced dataset, SMOTE, Machine Learning

1 Introduction

Child mortality rate, is defined as the probability of a child dying between birth and exactly 5 years of age (according to UNICEF). It is expressed as a number per 1000 live births. According to the estimates of the UN, this metric will get reduced to as low as 25 per 1000 live births by the end of this decade. Also, sometimes known as under-5 mortality rate or infant mortality rate, it is currently at 38 deaths per 1000 live births as of 2019. The vast majority of these deaths are happening in places with poor resources to monitor fetal health and alert on time.

1.1 Cardiotocography (CTG)

According to Wikipedia, a Cardiotocograph is an instrument used to monitor the fetal heartbeat and the uterine contractions during pregnancy and labour. An ultrasound transducer is placed on the mother's abdomen to do a continuous recording of the measurements, which sends sound pulses at ultrasound frequency and read the responses obtained. This is well explained in this article. [7]

It is crucial to properly read the CTG signal to correctly determine the health of a child. Some of the metrics that help with this are baseline rate, variability, accelerations, decelerations etc., Our aim for this case study is to use these features along with any extra feature engineering to predict the risk class of a given CTG profile.

2 Background

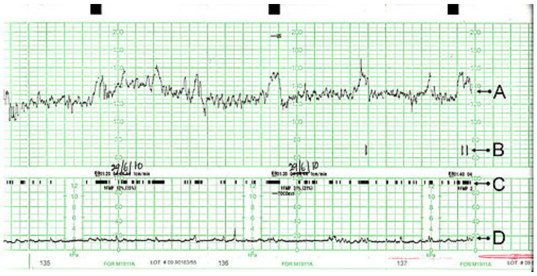

Fig. 1: CTG signal waveform. Image taken from Wikipedia. URL:- https://en.wikipedia.org/wiki/Cardiotocography

There is at least a couple of decades of research that went on for this particular problem. Back in 2000, *D. Ayres-de Campos et. al.,* [1] created a program named *Sisporto 2.0* that achieved a specificity of 99% along with a sensitivity of 100% in a preliminary study for the prediction of *poor neonatal outcome*. Utilizing the great advances in Machine Learning techniques in recent times, *A. Akbulut et. al.,* [2] experimented with predicting fetal anomaly status. They've compared the performance of several state-of-the-art binary classification algorithms such as Boosted Decision Trees, Locally Deep SVMs etc., With an accuracy of 89.5%, Decision Forest models outperformed the rest. As recent as 2019, *Z. Hoodbhoy et. al.,* [3] approached the same problem, but instead with an extra class. They've tried to classify the fetuses as Normal, Suspect and Pathological. Ten Machine Learning classification models were trained and tested. Unsurprisingly, the best performing models used ensemble techniques, with XGBoost giving the highest accuracy at 96% on training data and 92% on test data. This case study is heavily influenced by the above mentioned paper, in terms of the nature of the problem and the methodology used.

- *Nikhil Kumar Reddy Badveli is with the school of Computer Science and Electronic Engineering, University of Essex, Colchester, Essex, CO4 3SQ. E-mail: nb21979@essex.ac.uk*

3 Methodology

This case study aims to see if Machine Learning models are reliable and accurate enough to tackle this problem, so as to be able to use them in a real life setting in a low or middle income countries.

3.1 About the dataset

The dataset contains measurements of Fetal Heart Rate (FHR) and Uterine Contraction (UC) features on CTG of 2126 pregnant women, classified by expert obstetricians. It is obtained from the Machine Learning Repository maintained by University of California Irvine. Here's the link to the same:- https://archive.ics.uci.edu/ml/datasets/cardiotocography. There are a total of 22 columns in the data, out of which *fetal_health* column contains the output labels and the remaining 21 contains features obtained from CTG and those that are added using feature engineering techniques. The given data is highly imbalanced with 78% of the samples falling into Normal state, 14% falling into Suspect state and the remaining 8% into Pathological State. The same can be observed in the below figure.

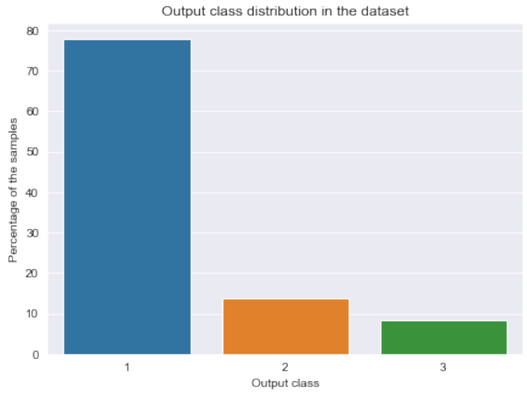

Fig. 2: Barchart showing data imbalance

This is not good for a classification problem since any model that predicts Normal state always, will be right about 78% of the time. And falsely classifying a child in Pathological state as Normal is extremely dangerous to the life of the fetus. So, our model should have as few *False Negatives* as possible in this case, ideally zero.

3.2 SMOTE to handle imbalanced data

There are multiple ways to handle imbalanced data. Refer to this article [4] to know more. Two of the primary methods are - oversampling and undersampling. Briefly speaking, in oversampling we try to increase the sample size of the minority classes so that all the classes have same number of samples. Conversely, in undersampling we delete samples from the majority class until the desired class distribution is achieved. For this case study, we employ Synthetic Minority Oversampling Technique (SMOTE), a simple and effective oversampling technique to boost up the sample sizes of *Suspect and Pathological classes*. Before proceeding further, we split the given dataset into train, test and validation sets with 80%, 10% and 10% of the total samples respectively. These sets are used as their name suggests.

3.3 Scaling the input data

One last step in the preprocessing of the data is to properly scale the input values so that the models can fit better. For this purpose, we use *StandardScaler* from the scikit-learn package, which scales the given input down to zero mean and one standard deviation.

3.4 Different models considered

The problem at hand can be considered as a classic multi-class classification problem for which multiple Machine learning algorithms exist. We test with the five most popular ones namely - Support Vector Machines (SVM), K Nearest Neighbors (kNN), Naive-Bayes classifier, Random Forest and Gradient Boosting algorithm. And then compare the results before and after oversampling along with the differences in performance among these five models. This is summarized in the Table 1 in the results section below. kNN can be thought of as a baseline classifier against which the rest of the models can be compared. Naturally, being ensemble techniques Random Forest and Gradient Boosting classifiers are expected to outperform the rest and it is observed to be so. And the primary assumption in Naive-Bayes classifier that the input features are independent and identically distributed (i.i.d) fails to apply in our case and it is expected to underperform the rest.

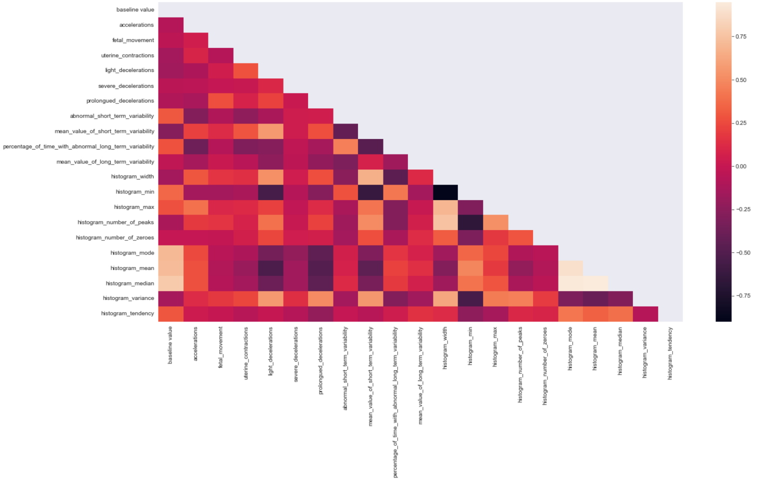

Fig. 3: Correlation heatmap between different features in the input data

3.5 Evaluation methods and metrics

It is rather important to choose a good set of performance evaluation metrics or else there's a decent chance of misinterpreting the results. We use classification accuracy, precision, recall, F1-score and finally, the number of False negatives in the case of *Pathological state*. It is out of the scope for this paper to elaborate on the definitions of these metrics. A good resource that explains the need of alternative metrics and how to choose them, is explained here. [5] We also utilize K-fold cross-validation as a technique to see if the models are overfitting to the training data and take any measures to overcome such scenario.

4 Results

The results section is organized into two subsections - one to compare and contrast the different models before and after oversampling, and the other to delve deep into the results of the best performing model out of the five.

4.1 Accuracy before and after SMOTE

TABLE 1: Classification accuracy for different models

Model	Before SMOTE (%)	After SMOTE (%)
SVM	90.17	93.84
kNN	89.98	95.79
Naive-Bayes	74.29	75.22
Random Forest	93.76	97.57
Gradient Boosting	95.84	96.68

As can be seen from the above table, all the models showed significant performance improvement after oversampling. Naive-Bayes classifier in particular stayed at the bottom even after SMOTE, for the reasons stated in the methodology section above. With an incredible above 96% classification accuracy both Random Forest and Gradient Boosting clearly stood out. Although it is not shown here, it is not just the accuracy that got improved after oversampling, all the other evaluation metrics increased as well to acceptable levels.

4.2 Summary of Random Forest Classifier results

TABLE 2: Confusion matrix for the Random Forest classifier. T. means True class.

	T. Normal	T. Suspect	T. Pathological
Pred. Normal	382	31	2
Pred. Suspect	6	381	6
Pred. Pathological	2	5	420

At a near perfect accuracy (approx. 98%), Random Forest classifier is the best performing model among all the models tested. And with the help of confusion matrix it is observed that there were zero cases of predictions that were falsely classified as *Normal* when the true case is *Pathological*. Also observed is the F1-score metric that shows the balance between Precision and Recall, which is estimated to be around 0.99 for the high risk category. So, it is safe to say the model gives reliable predictions.

TABLE 3: Classification report for Random Forest Classifier.

	Precision	Recall	F1-Score
Normal	0.99	0.95	0.97
Suspect	0.95	0.98	0.97
Pathological	0.99	0.99	0.99

4.3 Feature Importance

As explained well in this article [6], *"Feature importance refers to techniques that calculate a score for all the input features for a given model — the scores simply represent the 'importance' of each feature"*. For the purpose of this case study, we didn't manually create a function for this, instead we used the built-in *modelḟeature_importances_* attribute from the good old scikit-learn package. As can be observed from the Figure 4the top 2 features in terms of importance are related to variability - *abnormal_short_term_variability* and *percentage_of_time_with_abnormal_long_term_variability* and they've similar scores as well.

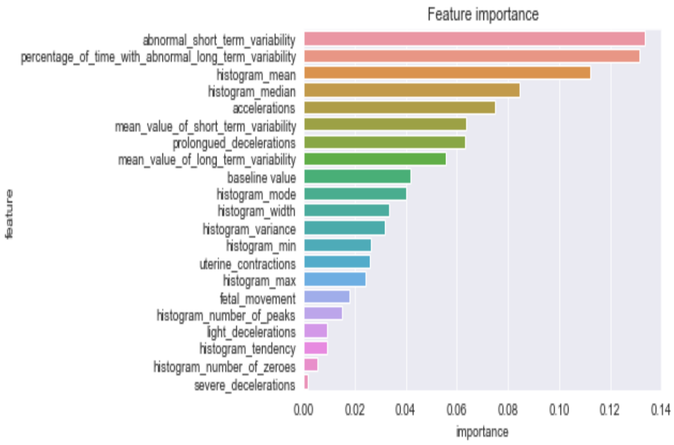

Fig. 4: Feature importance for the Random Forest Classifier

5 Discussion and Conclusion

We started off with the question of whether Machine Learning techniques can be reliably used in real-life setting and if the results from the above section are any indication, we can conclude the answer to be a resounding *yes*. It is worthy to note that these models are just a tool to give an initial screening of the fetal health and not an expert opinion. This case study has shown the importance of recognizing and properly handling imbalanced data. For any classification problem, this should be one of the first checks before proceeding to the modelling stage, since blindly training a model will lead to false conclusions. Although no feature selection technique is used in this case study due to smaller feature set, it is one of the important steps not to be forgotten. Alternatively, dimensionality reduction techniques such as Principal Component Analysis (PCA) could be employed to work with a smaller set of features. One of the things not mentioned in the methodology section, is about the sparse nature of the given dataset, in the sense that there are a lot of zeros. This could've been due to someone filling the missing values with zeros as part of the data cleaning process.

References

[1] D. Ayres-de Campos, J. Bernardes, A. Garrido, J. Marques-de Sa, and L. Pereira-Leite, "Sisporto 2.0: a program for automated analysis of cardiotocograms," *Journal of Maternal-Fetal Medicine*, vol. 9, no. 5, pp. 311–318, 2000.

[2] A. Akbulut, E. Ertugrul, and V. Topcu, "Fetal health status prediction based on maternal clinical history using machine learning techniques," *Computer methods and programs in biomedicine*, vol. 163, pp. 87–100, 2018.

[3] Z. Hoodbhoy, M. Noman, A. Shafique, A. Nasim, D. Chowdhury, and B. Hasan, "Use of machine learning algorithms for prediction of fetal risk using cardiotocographic data," *International Journal of Applied and Basic Medical Research*, vol. 9, no. 4, p. 226, 2019.

[4] 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset by **Jason Brownlee** from *Machine Learning Mastery* blog. URL:- https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

[5] Tour of Evaluation Metrics for Imbalanced Classification **Jason Brownlee** from *Machine Learning Mastery* blog. URL:- https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/

[6] Understanding Feature Importance and How to Implement it in Python by **Terence Shin** from *Towards Data Science* blog. URL:- https://towardsdatascience.com/understanding-feature-importance-and-how-to-implement-it-in-python-ff0287b20285

[7] How to Read a CTG by **Dr Lewis Potter** from *Geeky Medics* blog. URL:- https://geekymedics.com/how-to-read-a-ctg/