Springboard – DCS

Capstone Project 2

Predicting Fetal Cardiac Health Outcomes Using Cardiotocogram Data

By Michael Bobal

January 2023

1) Introduction

According to the Centers for Disease Control (CDC), in the United States, congenital heart defects (CHDs) are the most common types of birth defects. CHDs affect nearly 1% of children born, totaling nearly 40,000 cases per year. Of those 1% of children born with CHDs, about 25% have a critical CHD. Infants with critical CHDs will need surgery within their first year of life in order to prevent unnecessary death. Children in the fetal stage of development can also suffer from in-utero hypoxia - and even death - if certain fetal cardiac issues are not detected in time.

Cardiotocography (CTG) is a non-invasive medical test used to assess fetal heart rhythm, as well as uterine contractions in the mother. Obstetricians use information acquired from CTG to assess fetal health, make determinations about the necessity of preventative C-sections, and inform parents about possible health issues which may require surgical intervention for the child.

Currently, highly-trained physicians are required for the reading of CTG data, where they determine if the fetus' CTG reading class (a.k.a. Fetal CTG Class) is normal, suspect, or pathological (N, S, or P, respectively). Having access to a fast, high-fidelity classification system based on this CTG data would be an ideal scenario.

An accurate diagnostic determination is a critical step for our stakeholders: parents, obstetricians, surgeons, and hospitals. These parties must be vigilant in case the need arises for medical intervention (C-section to save the life of the fetus, possible cardiac surgery in the first year of life after birth, etc.). Therefore, the development of such a system was the goal of this project.

The result of our efforts have concluded that the data supplied to the data science team is insufficient. The raw data was poorly configured, confusingly explained, and contained repeat and time-overlapping entries of the same patients, which made leakage-free modeling impossible.

The notebooks detailing the process which led to our conclusions can be found at the following link:

https://github.com/mikebobal/springboard/tree/main/Capstone2/Fetal%20health%20idea

(2) Approach

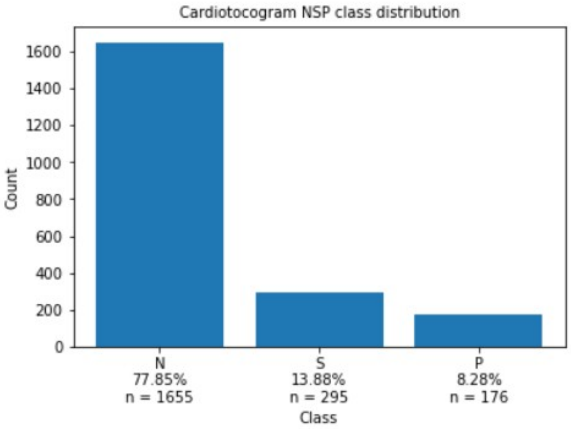
(2.1) Data Acquisition and Wrangling

The data came to us looking fairly clean. We found it in the University of California Irvine Machine Learning Repository. There were no missing values, but there were extra values due to some aggregate value rows. There were also many unneeded columns (evidenced by their exclusion from the “variables” list provided by the data documentation), which we removed. This cleaned dataset was saved for the next step.

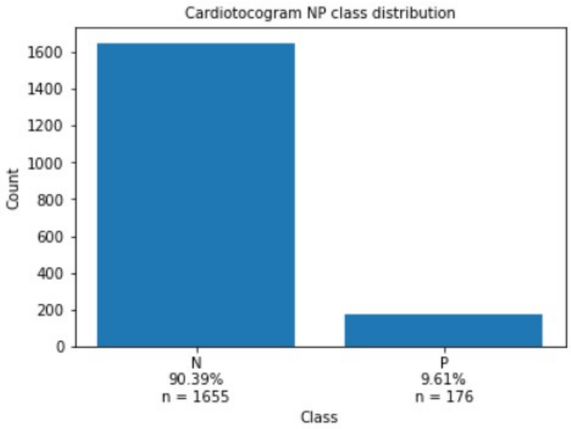
(2.2) Storytelling and Inferential Statistics

The data files presented us with 2126 CTG instances. Given the general reliability of the data source and the fact that the originating research was peer reviewed and published, we understood this to mean that the study used 2126 unique patients to compile their data. At the start of this project, we did not have access to the full research paper, and the researchers who conducted the study would not respond to our attempted communications. The study at hand was conducted in Portugal over 20 years ago, so we decided not to wait for a response that would never come.

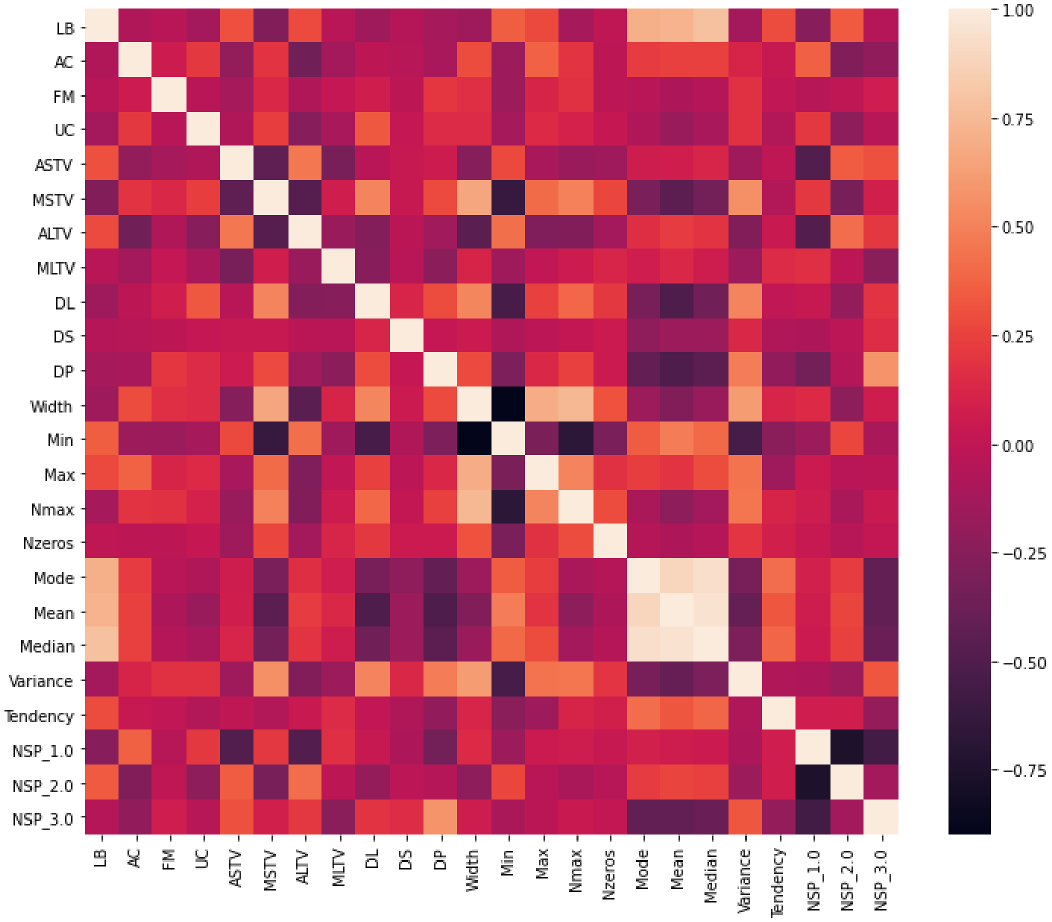
We discovered from the outset that the data was highly imbalanced on the diagnostic class. The researchers had medical experts classify each CTG instance as Normal, Suspect, or Pathological.



After considering the goal of our project, the decision was made to eliminate the Suspect class. In the health field, it is important to know if a patient has a serious health issue. Giving the diagnosis of “suspect” serves no cogent purpose. At this juncture, the plan was to use our best model at the end of the project to reclassify each Suspect class instance as either Normal or Pathological.



Afterward, we visualized how the features correlated with each other and with our target.

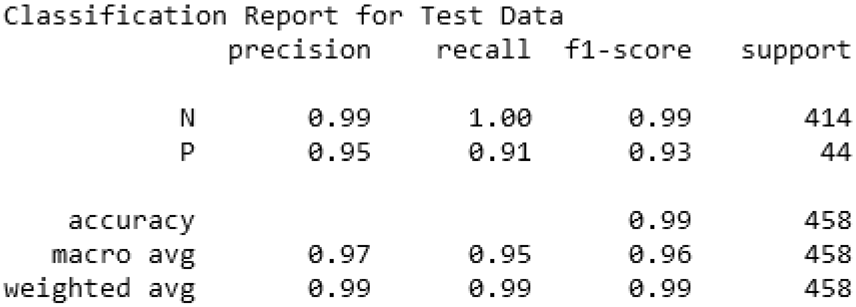


There were a few highly correlated variables seen. Particularly, Mean, Median, and Mode (all aggregate measures of central tendency).

(2.3.a) Baseline Modeling

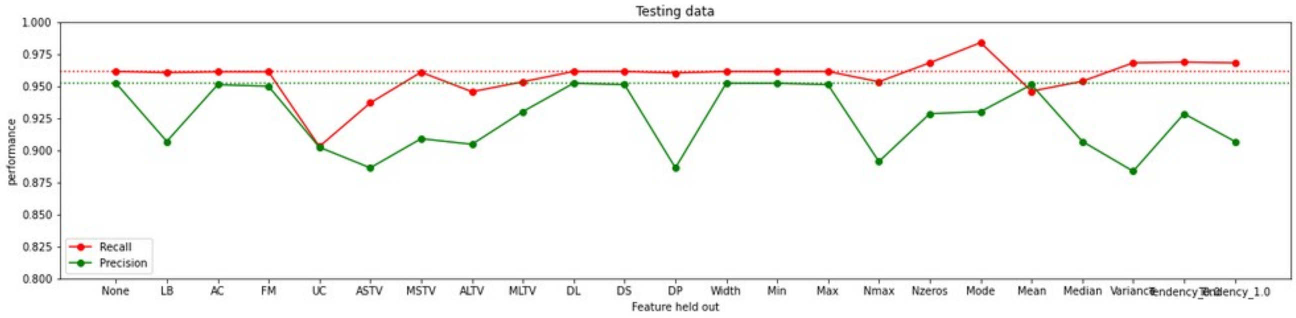
Due to our project being a classification problem, we began baseline modeling with a simple Logistic Regression. The performance measure we decided to use was recall. Medical classification problems tend to value reducing false negatives. It is better to err on the side of telling a healthy patient that they have a bad prognosis, compared to telling a seriously unhealthy patient that they are fine.

Unfortunately, it was immediately clear that there was some sort of data issue. The simplest model, without the benefit of hyper-parameter tuning, was able to achieve a positive class recall score of 0.91.



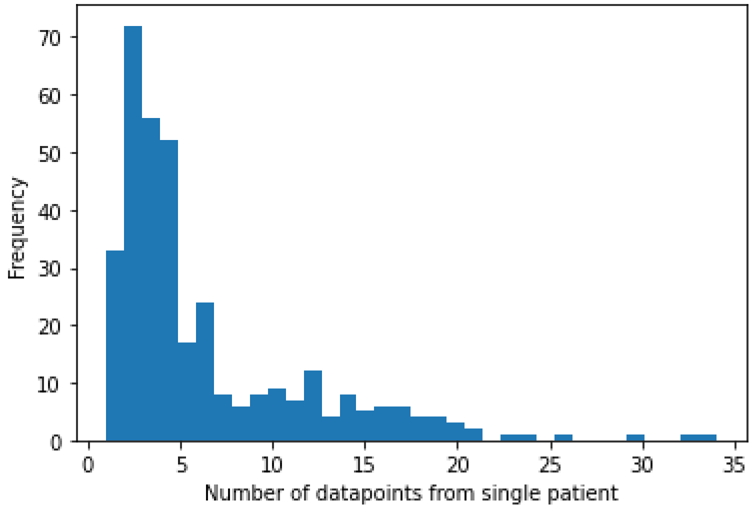
```
Classification Report for Test Data
              precision    recall  f1-score   support

           N       0.99      1.00      0.99       414
           P       0.95      0.91      0.93        44

    accuracy                           0.99       458
   macro avg       0.97      0.95      0.96       458
weighted avg       0.99      0.99      0.99       458
```

In order to investigate whether we could be experiencing data leakage from any particular variable, we established a one-out function which performed the modeling repeatedly while holding one feature out each time. Charting the precision and recall results could give insight as to whether there is data leakage from one of our variables.



The fluctuations in the hold-out results were not out of the ordinary for a dataset, so we determined that there was not a singular feature leaking data.

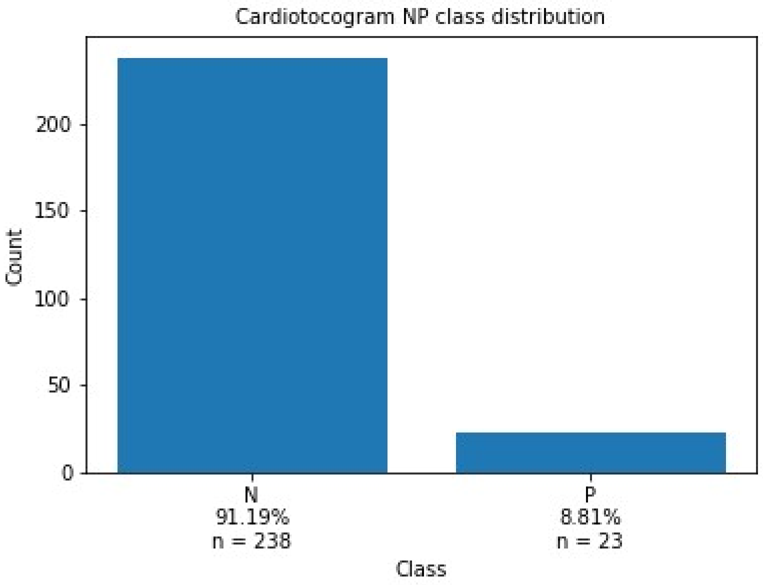
Next, we needed to investigate whether there was a more generalized data leakage issue. After investigating the subjects from which the data-points were derived, something interesting and vital for creating an accurate and leakage-free model was revealed. Even though we have 2126 data-points, there are actually only 352 unique patient identifiers. The data source was not clear about this quirk of the data. Those rows of data added up to 2126 because each patient had multiple CTG tests, and therefore was sampled from multiple times. This was the cause of our leakage.



While most patients had 5 or fewer tests, some patients had upward of 30 CTG examinations. Any occurrence of a CTG from one patient's being included in the training set would mean over-fitting to the particular patients included in the data, and that the test set is an unreliable data source for measuring model performance.

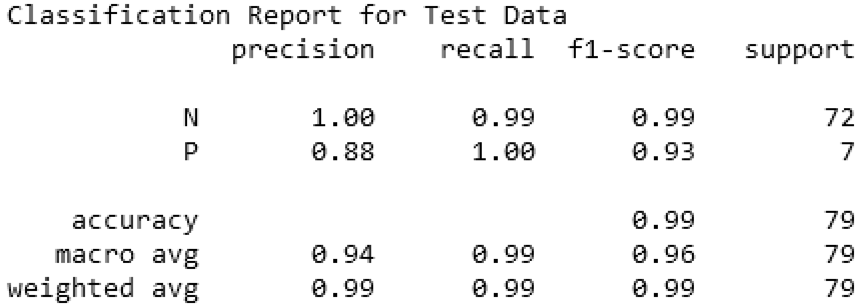
(2.3.b) Baseline Modeling (after data leakage discovery)

We decided to redo portions of the preprocessing step, taking measures to correct for quasi-repeat data points. In order to not have patients repeated in the data, we aggregated each feature into one number using their mean. The NSP class was assigned ordinal labels 1,2,3 for N,S,P, respectively, and aggregated using that mean, then rounded to the nearest whole number for final categorization. While imperfect, we deemed this step essential while dealing with such a severe leakage issue.



What we were left with data-wise was a very small number of positive samples (23), while maintaining the approximately 10:1 Normal:Pathological ratio from before.

Unfortunately, again, our baseline modeling was “too good to be true”.



```
Classification Report for Test Data
             precision    recall  f1-score   support

           N       1.00      0.99      0.99        72
           P       0.88      1.00      0.93         7

    accuracy                           0.99        79
   macro avg       0.94      0.99      0.96        79
weighted avg       0.99      0.99      0.99        79
```

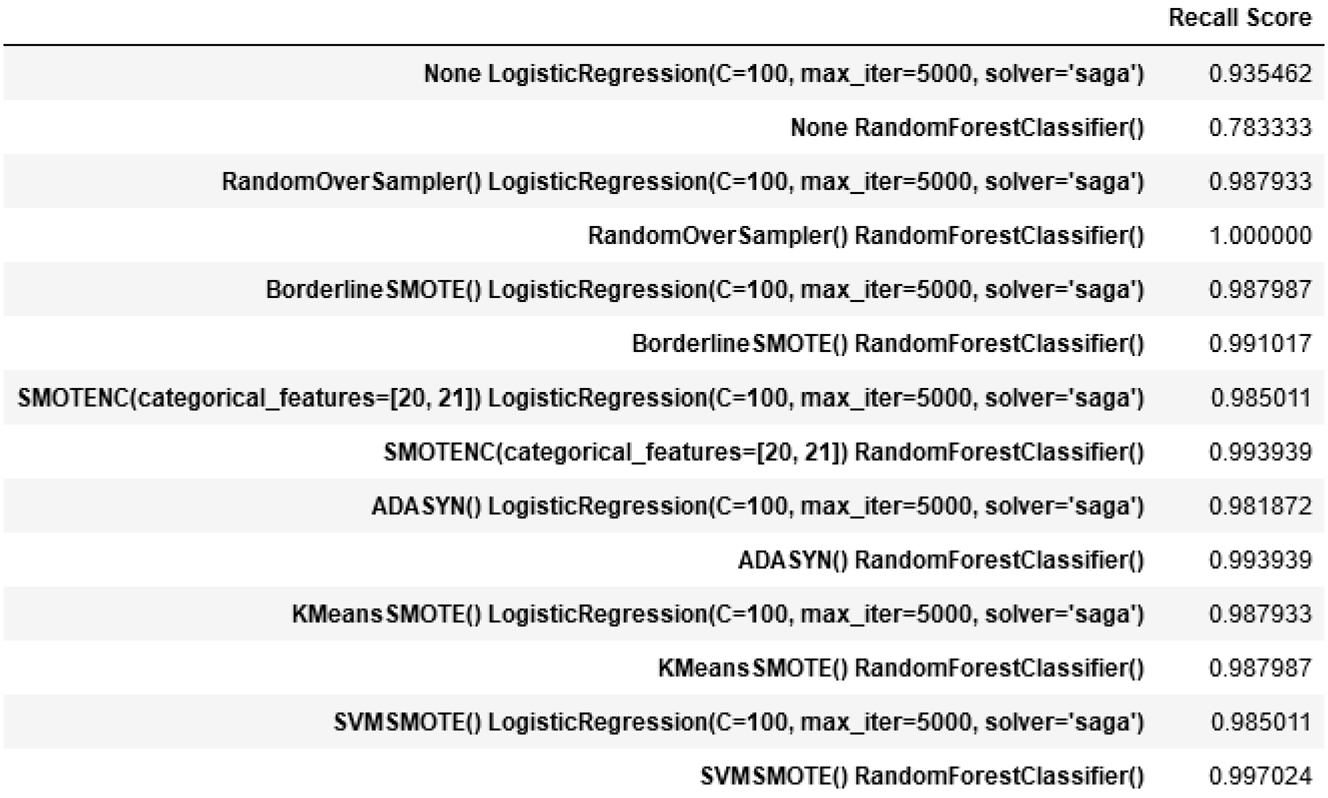
We saw an otherworldly 1.00 positive class recall score on a simple model built from a logistic regression algorithm.

With only 7 Pathological class samples to test on, we set aside our continued suspicions regarding the improbable performance of a simple model on the data for a short time. Due to the small size of the overall dataset, and the small size of the positive class, we wanted to see how the model would react to the use of oversampling techniques on the data.

(2.4) Extended Modeling

Of the oversampling methods available, we utilized RandomOverSampler and multiple varieties of Synthetic Minority Over-sampling Technique (SMOTE) such as BorderlineSMOTE, SMOTENC, ADASYN, KMeansSMOTE, SVMSMOTE.

This strategy was implemented via a function which looped through the different oversampling techniques. We tested the performance of models fit to that over-sampled data using stratified k-fold and cross validation (leaving alone the smaller test split). We utilized basic Logistic Regression and Random Forest Classifier algorithms.



	Recall Score
None LogisticRegression(C=100, max_iter=5000, solver='saga')	0.935462
None RandomForestClassifier()	0.783333
RandomOverSampler() LogisticRegression(C=100, max_iter=5000, solver='saga')	0.987933
RandomOverSampler() RandomForestClassifier()	1.000000
BorderlineSMOTE() LogisticRegression(C=100, max_iter=5000, solver='saga')	0.987987
BorderlineSMOTE() RandomForestClassifier()	0.991017
SMOTENC(categorical_features=[20, 21]) LogisticRegression(C=100, max_iter=5000, solver='saga')	0.985011
SMOTENC(categorical_features=[20, 21]) RandomForestClassifier()	0.993939
ADASYN() LogisticRegression(C=100, max_iter=5000, solver='saga')	0.981872
ADASYN() RandomForestClassifier()	0.993939
KMeansSMOTE() LogisticRegression(C=100, max_iter=5000, solver='saga')	0.987933
KMeansSMOTE() RandomForestClassifier()	0.987987
SVMSMOTE() LogisticRegression(C=100, max_iter=5000, solver='saga')	0.985011
SVMSMOTE() RandomForestClassifier()	0.997024

Again, the results are astoundingly good. Some of the models, without even the benefit of hyper-parameter tuning, are able to achieve perfect recall — an extremely unlikely outcome.

(3) Findings

Unfortunately, upon further inspection of the raw data, we discovered:

1. The CTG entries corresponding to each patient overlapped in time. Some of the data-points encompassed entirely the rest of that patient's data.

2. Some unseen data leakage is continuing to occur, even after the fixes we established.

As a fellow published author of research, and without intending to denigrate the work done by these individuals, I would make this assessment: the data collection process was not on par with scientific standards. If they were introducing an experimental variable (like a treatment) at different times, or measuring how vital signs changed over time, then having multiple snapshot measurements of the same patient repeatedly would make sense. However, the intention of this data was purely to create a diagnostic model to relieve the need for highly trained medical professionals doing the same work. In fact, some of the patients had their classification change over time, which would only serve to confuse the modeling step (but many instances of this were removed prior to modeling due to their aggregated class determination being found as “Suspect” – the eliminated class).

The most confusing aspect of this scenario is this dataset's inclusion in the UCI Machine Learning Repository. The data is presented as 2126 unique data-points ripe for machine learning classification algorithms. However, the real number of actionable unique data-points is a mere fraction of that, and the way it was collected makes impossible the interpretation of the data values presented.

(4) Conclusions and Recommendations for the Clients

In conclusion, we tried to overcome a poorly constructed dataset by reducing the obvious data leakage. After attempting to model the fixed data, we still found the results to be too exemplary to be real. Unfortunately, for this reason, this project needed to be discontinued without a fulfilling conclusion.

The conclusions of this project are disappointing, but do occur. In order to try this project again, the data team would need the implementation of a strict, simple, standard process for data collection and entry.

Based on our findings, we strongly recommend making the following concrete data collection improvements:

1. Supply the data team with CTG data from more subjects. Increasing the testing is a more reliable approach than artificially oversampling limited data.

2. Include in future data collection only a single data entry per patient. This was a cause of major information leakage, led to over-fitting during modeling, and is a bad practice for the type of problem that we are trying to solve.

3. Ensure a standard amount of time that each CTG may collect data for.

Once we have obtained properly collected and organized data, we can re-commence our attempt to provide a very accurate system for identifying problematic CTG results in lieu of medical experts being present.

(5) Consulted Resources

- Data source: https://archive.ics.uci.edu/ml/datasets/cardiotocography

- Originator of the data: Ayres de Campos et al. (2000) SisPorto 2.0 A Program for Automated Analysis of Cardiotocograms. J Matern Fetal Med 5:311-318.

- Mentorship and guidance: AJ Sanchez