Springboard – DCS

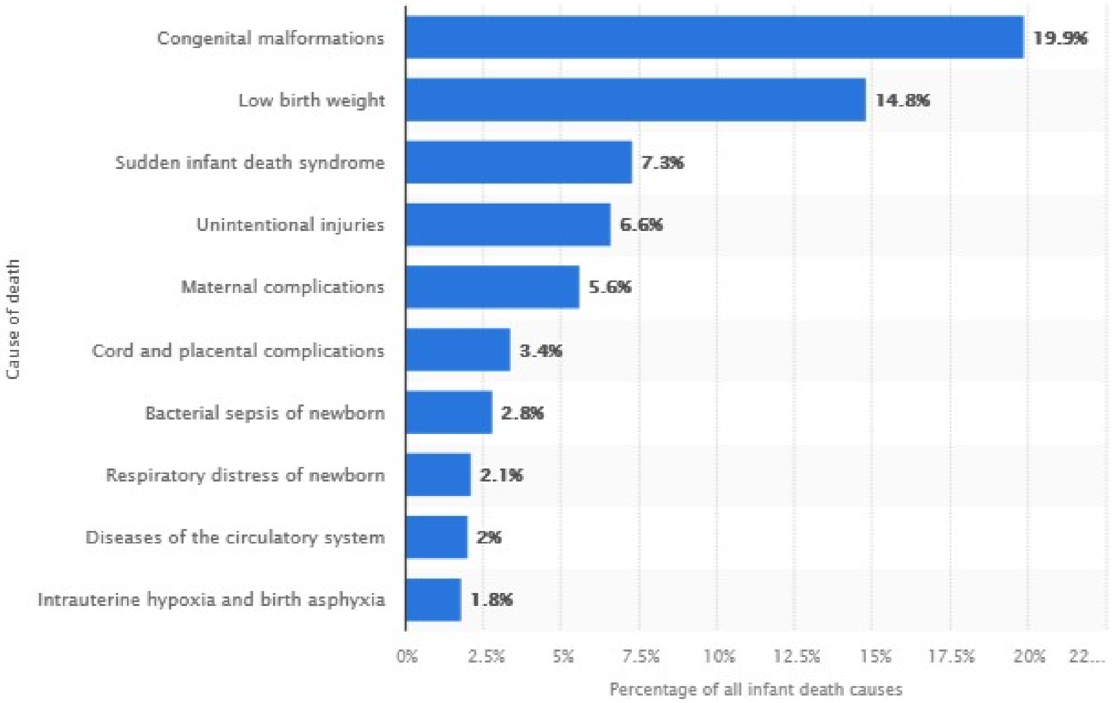
Capstone Project 2

Predicting Fetal Cardiac Health Outcomes Using Cardiotocogram Data

By Michael Bobal

January 2023

Introduction



- Congenital malformations like heart defects (CHD), are responsible for nearly 20% of infant deaths.

- CHDs affect nearly 1% of children born alive, totaling nearly 40,000 cases per year.

- Of those 1% of children born with CHDs, about 25% are critical.

https://www.statista.com/statistics/946675/leading-causes-of-infant-death-in-the-us/

Introduction

Cardiotogography (CTG) is a non-invasive, in-utero fetal heart-health test that obstetricians use to detect the presence of CHD.

Interpreting the results usually requires a highly trained physician, who carefully considers the multiple measurements of the CTG in order to classify fetuses as either normal, suspect, or pathological.

Accurate diagnosis is a critical step for our stakeholders:
Affected children, parents, obstetricians, surgeons, and hospitals.

Goal:

Develop a machine learning model that uses CTG data to quickly and accurately predict the cardiac status of a fetus.

Approach: Data Acquisition and Wrangling

Data is from University of California Irvine Machine Learning Repository

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

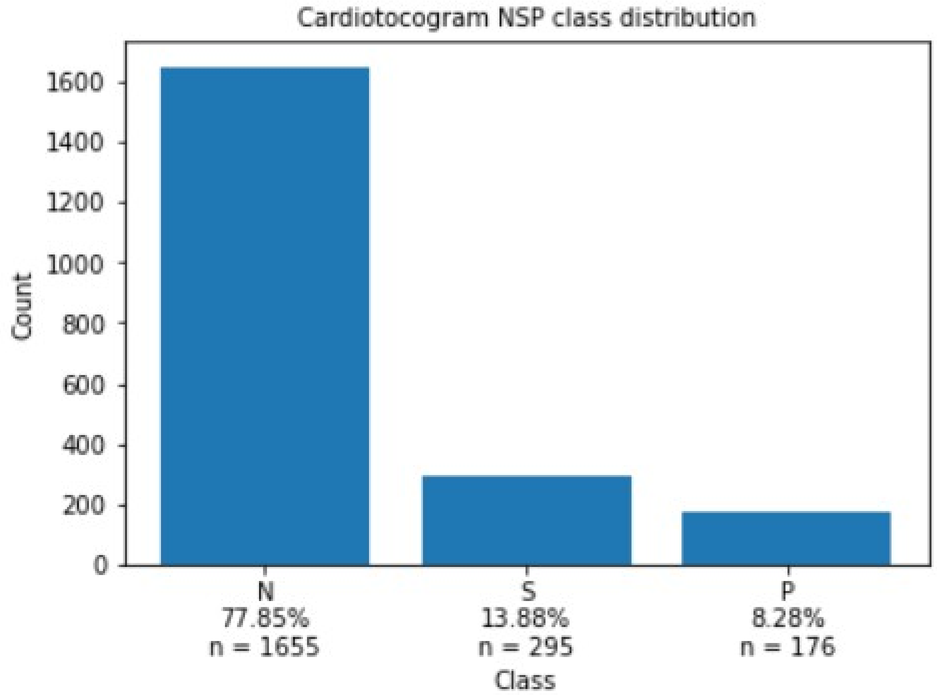
Our project only used the columns corresponding to features suggested in the data file, totaling 21 independent variables for 2126 CTG instances

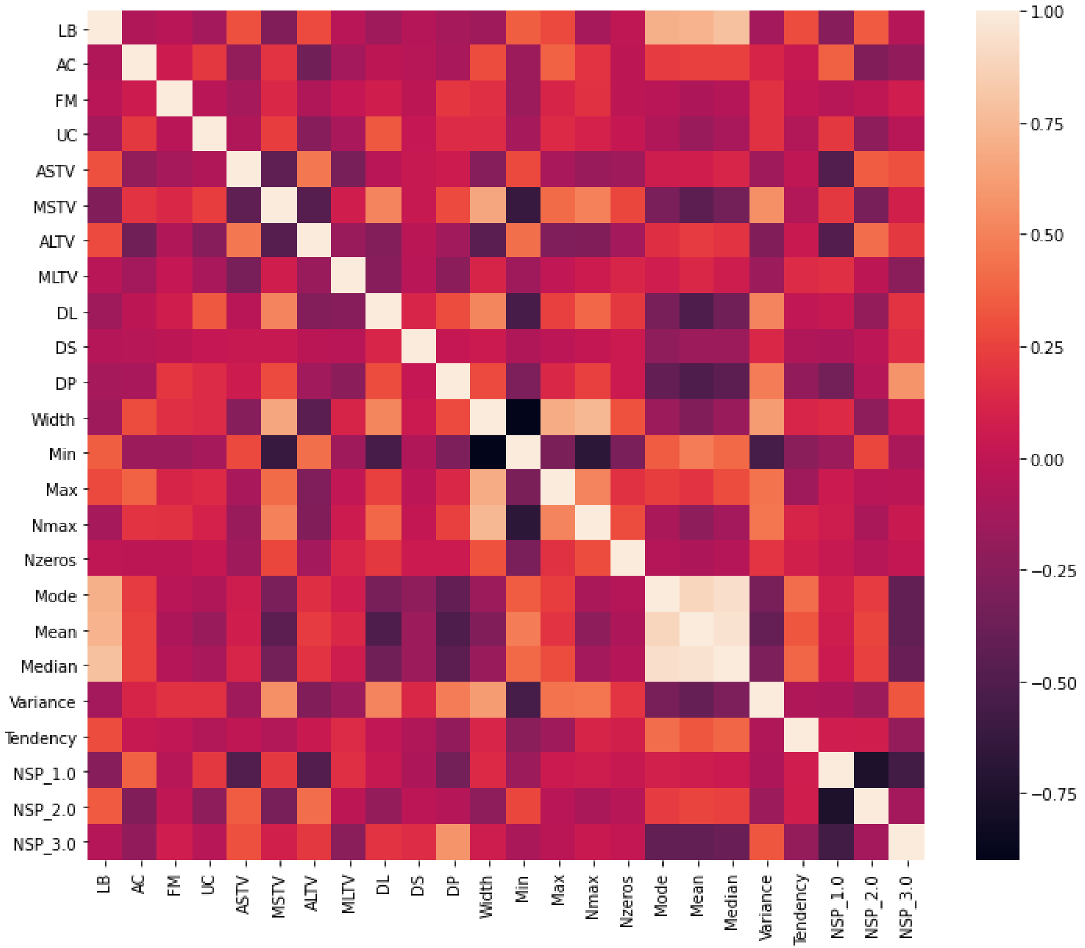
Our data:

- Each CTG instance classified as Normal, Suspect, or Pathological by a medical professional

- Includes “Suspect” as a class designation, which we eliminated from modeling

- Highly imbalanced





Feature correlations:

- Highly correlated features with features:
  >Mean, Median, Mode
  >Min, Width

- Highly correlated features with target:
  >ASTV and ALTV with Normal class
  >Mean/Median/Mode with Pathological class

Baseline Modeling

Goal of modeling: High sensitivity (good detection abilities)

Performance metric of choice: Recall (True Positives / All Positives)

Reason: Medical classification problems tend to value reducing false negatives

Good is Bad

- Initial algorithm chosen for model-building was Logistic Regression

- Model performance without any tuning achieved improbable results

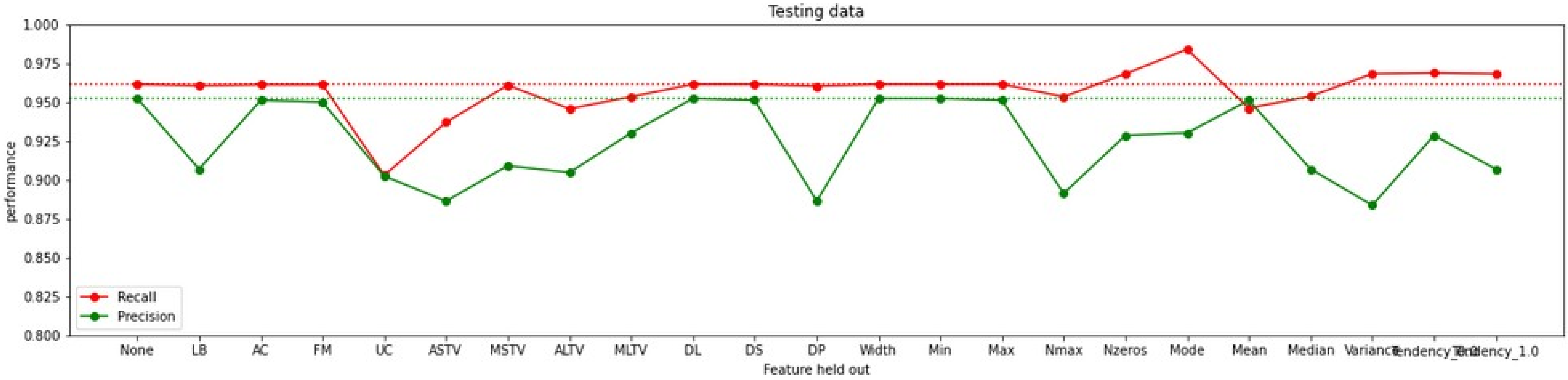
- >90% on all standard performance metrics

- Recall score = 0.91

- Data leakage is a likely culprit

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



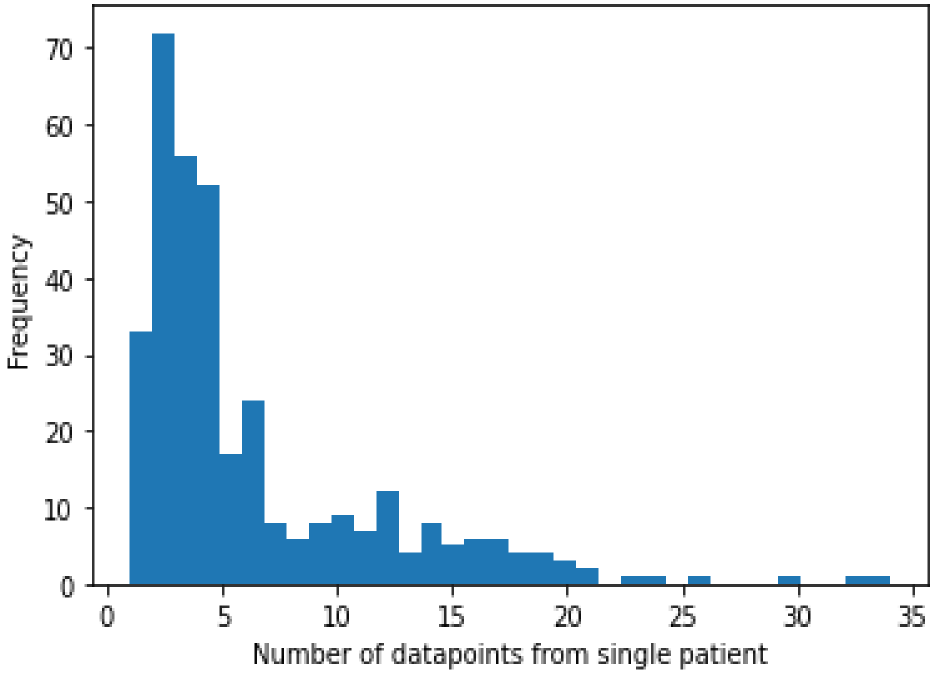
<u>Investigating single features for leakage</u>

- Using a feature hold-out function, the model was repeatedly run.

- Precision and Recall scores for each iteration were charted.

- No obvious signs that one feature is the source of the data leakage issue.

Leakage Located



- Further review of the data showed that most subjects had multiple CTG data entries

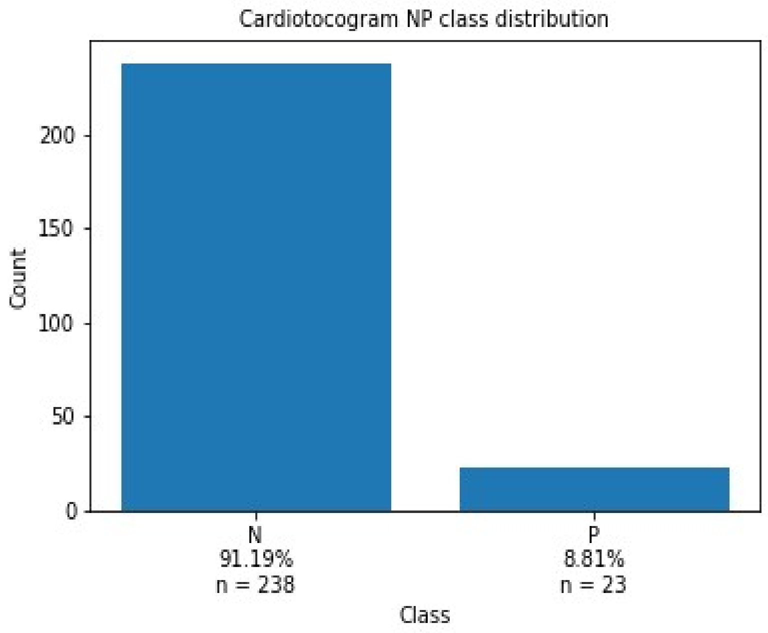
- Data from subjects was ending up in both the training set and the testing set for modeling

- The model was over-fit to the particular patients involved

Redo EDA

- Using grouping and aggregation, we condensed the data from 2126 to 352 data-points

- Class imbalance remained ~10:1



Redo Baseline Modeling

- Repeating our steps from before, we used Logistic Regression

- Again, the results are astoundingly good. Too good.

- Recall = 1.00

- The leakage issue is not solved

```
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
```

The problem of leakage is consistent:

Comparing cross-validation performance when using over-sampling techniques on the data show similarly excellent results.

	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

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.

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.

Conclusions

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.

Thank you to AJ, my mentor,
for helping to steer me in the proper direction throughout this project.