Rick Stephenson Capstone Project 3 -- final report

Classify fetal health in order to prevent child and maternal mortality

Project inspired by and data downloaded from Kaggle Dataset:
https://www.kaggle.com/andrewmvd/fetal-health-classification

Those data came from:

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

https://onlinelibrary.wiley.com/doi/10.1002/1520-6661(200009/10)9:5%3C311::AID-MFM12%3E3.0.CO;2-9

From the Kaggle file:

> *Reduction of child mortality is reflected in several of the United Nations' Sustainable Development Goals and is a key indicator of human progress.*
> *The UN expects that by 2030, countries end preventable deaths of newborns and children under 5 years of age, with all countries aiming to reduce under-5 mortality to at least as low as 25 per 1,000 live births.*
>
> *Parallel to notion of child mortality is of course maternal mortality, which accounts for 295 000 deaths during and following pregnancy and childbirth (as of 2017). The vast majority of these deaths (94%) occurred in low-resource settings, and most could have been prevented.*
>
> *In light of what was mentioned above, Cardiotocograms (CTGs) are a simple and cost accessible option to assess fetal health, allowing healthcare professionals to take action in order to prevent child and maternal mortality. The equipment itself works by sending ultrasound pulses and reading its response, thus shedding light on fetal heart rate (FHR), fetal movements, uterine contractions and more.*

The criterion for success for this project was a working model that correctly identifies unhealthy babies. The bias is towards reducing false negatives (calling unhealthy babies healthy) so that interventions will be performed when needed. Using the provided dataset, a classification model was created to take the various readings from the Cardiotocograms and use them to predict fetal health.

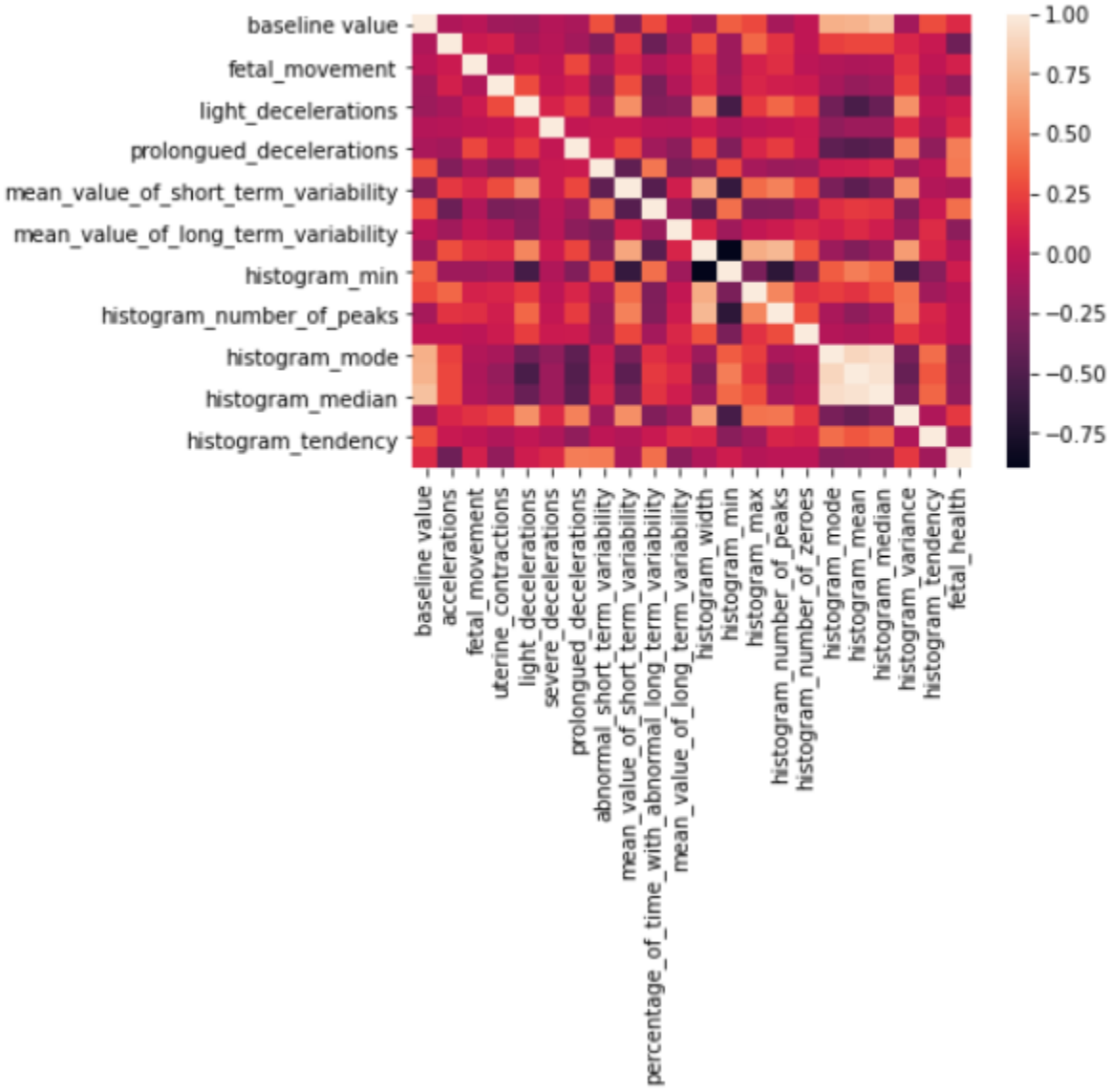
The data were originally presented in a comma separated text file containing 22 columns of data. The last column in the table, named “fetal_heath” contained the expert coded evaluations:

1 (Normal), 2 (Suspect) and 3 (Pathological)

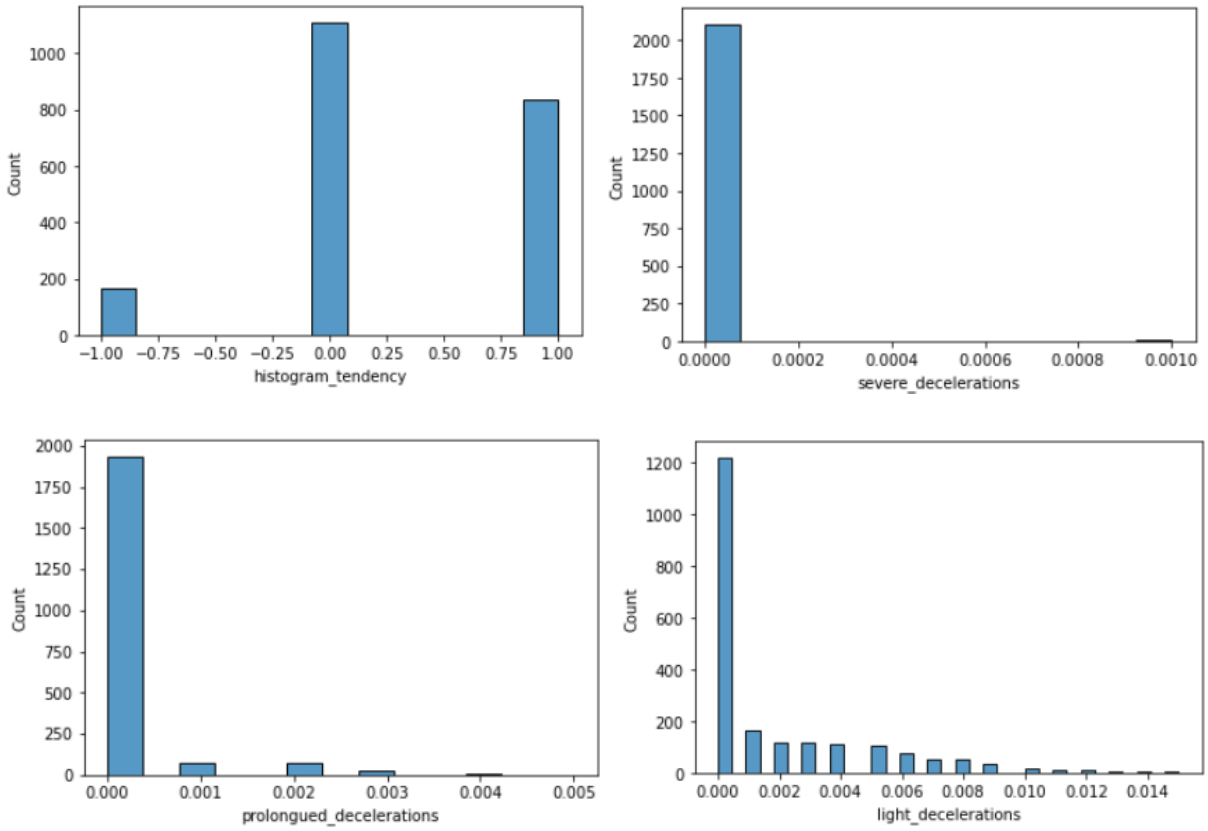
These were changed to 0 (Normal), 1 (Suspect) and 2 (Pathological) for the sake of the algorithm. The data were very clean, with no missing or null values and no apparent typos or

inappropriate entries. All columns consisted of solely floating point data. Thirteen (13) rows were found to contain duplicate information. These rows were discarded.

Heatmap of all column was generated to determine correlations between variables. No significant autocorrelation was observed which required further action.

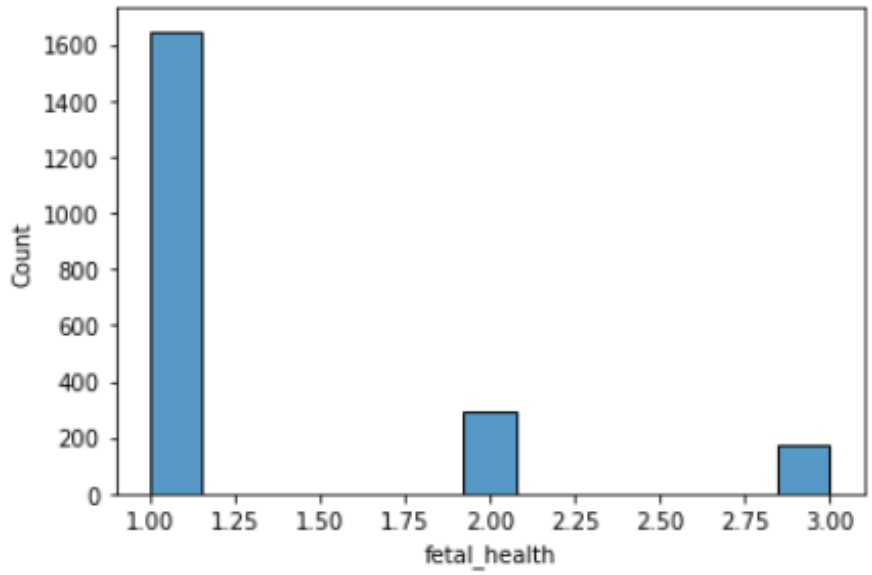


Examination of the distribution of values for some of the columns showed that the values were collected into only a few unique numbers.

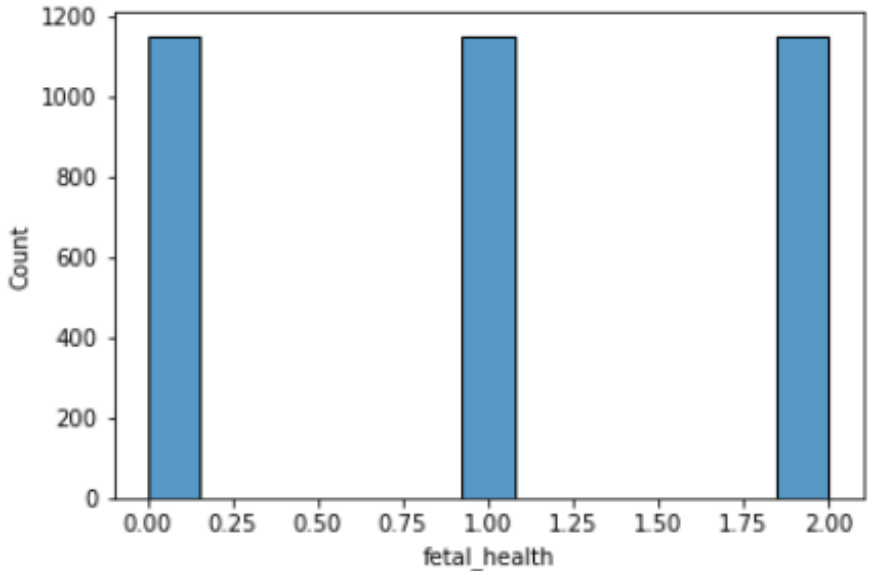


The “light_decelerations” column was left as is. Even though the data looks like they may be discretely distributed, there are enough unique values to be treated as continuous. The “histogram_tendency” and “prolonged_declerations” columns were one-hot encoded to produce 9 sparse columns coded as 1 or 0. Finally, the “severe_decelerations” column was found to have only two unique values, “0” and “.001”. Only 7 of the rows had “.001” as an entry. Six (6) of the 7 rows were associated with a Fetal health of “3”, or “pathological.” Since this may be strong indicator for a pathological case, this colum was left in the model. The 7 values were simply coded as “1” so that it could be treated as categorical.

Examining the output data shows that the outputs are unbalanced, with many more values in the “normal” category than the “pathological” category.



This necessitated that balancing be performed before the model was created. After the data were split into training and testing sets, Synthetic Minority Oversampling Technique (SMOTE) method was run. This technique creates new values in the minority class to balance the training data. Specifically, the SMOTE-NC function was used. This function allows the user to specify categorical columns. The function then keeps those columns as categorical when creating synthetic values. The data now contain a larger number of rows with balanced set of outputs. These input column were standardized and the models prepared.



Three different techniques were used to model the fetal health predictions. As a baseline, a K-nearest neighbors (KNN) model was created. The number of neighbors to use was determined using a cross validated grid search. The best predicted K was 1. The F1 score was used to compare each model's overall predictive power. The weighted average F1 score for the KNN model was only 0.90. This was used as a baseline for the other models.

	precision	recall	f1-score	support
0	0.96	0.95	0.95	494
1	0.70	0.74	0.72	88
2	0.83	0.87	0.85	52
accuracy			0.91	634
macro avg	0.83	0.85	0.84	634
weighted avg	0.91	0.91	0.91	634

KNN, K=1

Next, a Random Forest Classifier was tried. Again, cross validated grid search was performed to determine the optimal number of estimators for the Random Forest model. The best estimator value was found to be 40. This model gave a weighted average F1 score of 0.94. While better than KNN, this model still is not accurate enough for clinical use.

	precision	recall	f1-score	support
0	0.96	0.97	0.96	494
1	0.80	0.76	0.78	88
2	0.96	0.92	0.94	52
accuracy			0.94	634
macro avg	0.91	0.88	0.89	634
weighted avg	0.94	0.94	0.94	634

Random Forest, 40 Estimators

The XGBoost technique can often provide good results “out of the box.” This was demonstrated when an XGBoost model was created using default values that gave a weighted average F1 score of .95. Hyperparameters were then tuned, again using cross validated grid searches. The tuned parameters were “max depth”, “n estimators” and “learning rate.” Once those were defined, the “column sample by tree” and “gamma” parameters were tuned, but the results were unchanged. The final model yielded better individual F1 scores than the Random Forest Model and a weighted average F1 score of 0.95.

	precision	recall	f1-score	support
0	0.96	0.98	0.97	494
1	0.87	0.78	0.83	88
2	0.98	0.98	0.98	52
accuracy			0.95	634
macro avg	0.94	0.92	0.93	634
weighted avg	0.95	0.95	0.95	634

XGBoost, Tuned

The final XGBoost model can be used for clinical prediction with an F1 score of .98 for the most clinically relevant case and a weighted average F1 score of 0.95.

	KNN	Random Forest	Stock XGBoost	Tuned XGBoost
F1 - “normal”	.95	.96	.97	**.97**
F1- “suspect”	.72	.78	.80	**.83**
F1 - “Pathological”	.85	.94	.96	**.98**
weighted average F1	.91	.94	.95	**.95**

The final parameters of the XGBoost model are:

```
XGBClassifier(
        base_score=0.5,
        booster='gbtree',
        colsample_bylevel=1,
        colsample_bynode=1,
        colsample_bytree=1,
        gamma=0,
        gpu_id=-1,
        importance_type='gain',
        interaction_constraints='',
        learning_rate=0.2,
        max_delta_step=0,
        max_depth=6,
        min_child_weight=1,
        missing=nan,
        monotone_constraints='()',
        n_estimators=140,
        n_jobs=6,
        num_parallel_tree=1,
        objective='multi:softprob',
        random_state=42,
        reg_alpha=0,
        reg_lambda=1,
        scale_pos_weight=None,
        seed=42, subsample=1,
         tree_method='exact',
        use_label_encoder=False,
        validate_parameters=1,
        verbosity=None)
```

Further hyperparameter tuning could yield better results. Further study could be performed using another data set and comparing the two outputs.