Fetal Health Classification

Charrie Mascio

Fetal Health Classification

Purpose

The purpose of this research is to clean and prepare the fetal health classification data into a useable format to predict fetal health outcomes accurately using Cardiotocograms (CTGs) data to prevent or decrease child and maternal mortality.

A decrease in the child mortality rate is one of the United Nations' Sustainable Development Goals. Parallel to the notion of child mortality is of course maternal mortality, which accounts for 295 000 deaths during and following pregnancy and childbirth (as of 2017). Most of these deaths (94%) occurred in low-resource settings, and most could have been prevented.

CTGs can be used as a simple and cost-effective method for assessing fetal health and preventing child and maternal mortality. CTGs work by sending ultrasound pulses and reading their response to gather information about the fetal heart rate (FHR), fetal movements, uterine contractions, and other factors.

CTG is the most widely used in the clinical routine evaluation to detect fetal state and has enabled clinical practitioners to detect signs of fetal compromise at an early stage. It provides information on uterine respiration and fetal heart rate, which can be used to determine whether the fetus is healthy, suspect, or pathological.

Having a predictive model would be a valuable tool for the healthcare industry. This healthcare facility wants to make an impact by being able to predict fetal health outcomes which would in the end prevent or decrease child and maternal mortality.

Dataset:

The dataset[1] is from Kaggle and has 2,126 rows of 22 features extracted from CTG exams, which were then classified by three expert obstetricians into 3 classes: Normal, Suspect, and Pathological.

The data description is below:

- baseline_value: Baseline Fetal Heart Rate (FHR)
- accelerations: Number of accelerations per second
- fetal_movement: Number of fetal movements per second
- uterine_contractions: Number of uterine contractions per second
- light_decelerations: Number of LDs per second
- severe_decelerations: Number of SDs per second
- prolongued_decelerations: Number of PDs per second
- abnormal_short_term_variability: Percentage of time with abnormal short-term variability
- mean_value_of_short_term_variability: Mean value of short term variability
- percentage_of_time_with_abnormal_long_term_variability: Percentage of time with abnormal long-term variability
- mean_value_of_long_term_variability: Mean value of long term variability
- histogram_width: Width of the histogram made using all values from a record
- histogram_min: Histogram minimum value

[1] Ayres de Campos et al. (2000) SisPorto 2.0 A Program for Automated Analysis of Cardiotocograms. J Matern Fetal Med 5:311-318 (Link9:5%3C311::AID-MFM12%3E3.0.CO;2-9))

- histogram_max: Histogram maximum value
- histogram_number_of_peaks: Number of peaks in the exam histogram
- histogram_number_of_zeroes: Number of zeroes in the exam histogram
- histogram_mode: Histogram mode
- histogram_mean: Histogram mean
- histogram_median: Histogram median
- histogram_variance: Histogram variance
- histogram_tendency: Histogram trend
- fetal_health: Fetal health:
  - 1 - Normal
  - 2 - Suspect
  - 3 - Pathological

Data Wrangling

The purpose of Data Wrangling is to clean and prepare the fhc_dataset into a usable format to predict fetal health outcomes accurately using CTGs data to prevent or decrease child and maternal mortality.

The following tasks were performed. Starting with loading the chosen dataset:

Reviewed the fhc_dataset more in depth by becoming familiar with the different columns. Check the column names, shape, info, summary statistics for each of the columns and unique values of the dataset. Started cleaning the data for missing values or NaN values and duplicates. The data did not have any missing values, NaN's or any duplicates.

Looked into the value counts for each feature to understand the dataset and made a histogram for the data to see the distribution of features and see any useful values that may need further investigation.

There is further need for investigation and data analysis to reach a more concrete conclusion.

Exploratory Data Analysis

The following tasks were performed:

We started our EDA analysis by assessing the quality of the data. We do this to understand what information is in the data and make sure what the data stands for and what is missing.

We concluded from the data wrangling that we have no missing data and have a tidy data.

We generated a Pandas profiling report which shows us correlation tables, heatmaps which gives us an overview that some of the features might have strong correlation to each other.

The fetal_health column showed 1 to 3 which represents the three different classes which are: 1 for normal, 2 for suspect, and 3 for pathological which becomes the dependent variable to be predicted based on the other features. From its value counts, it looks like the normal class has most of the data samples.

Next, we looked at correlations of the different features by focusing on correlation coefficients between -1.0 and -0.5 or between 0.5 and 1.0. The top three listed are baseline_value, histogram_max and histogram_mean.

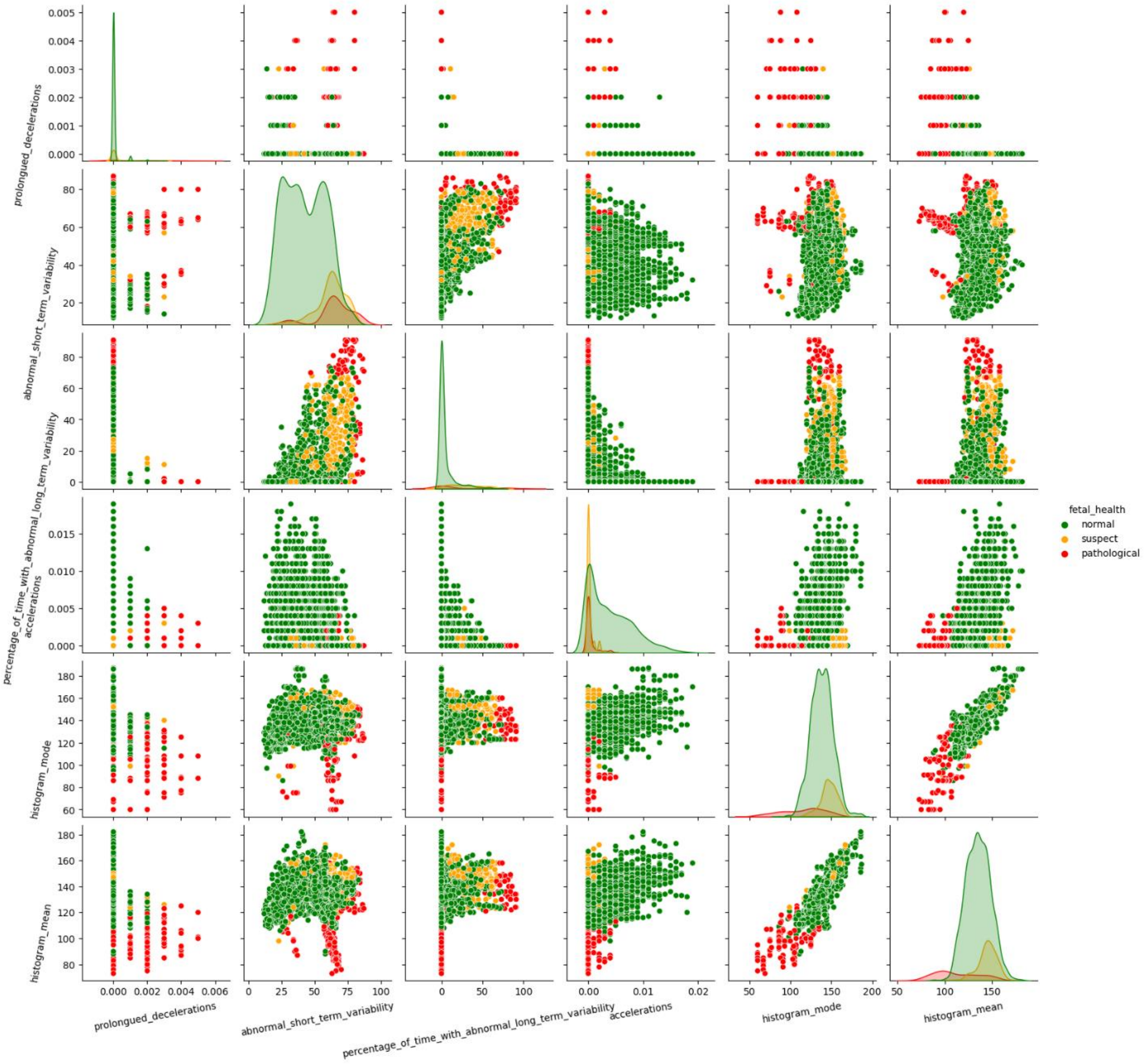
The variance_inflation_factor (VIF) was analyzed to check for collinearity between the different features and showed that there is a high possibility of collinearity between the features.

Other visualizations like the boxplot and heatmap were created and only more questions came up to see if the variables would show some more interaction. This leads to the need for further exploration.

Looked at the correlation of the different features against fetal_health and the top three that have the strongest correlation with fetal_health are:

- prolongued_decelerations
- abnormal_short_term_variability
- percentage_of_time_with_abnormal_long_term_variability

For visualization, the pair plots (below) showed insights into the relationships between the different features and fetal_health. We see that all the values are shifted towards the low end, especially for the pathological data. The normal data shifted toward more toward the higher end with the suspect data somewhere in the middle. We do see a strong correlation between histogram_mean and histogram_mode.



The best statistical tests to compare more than two groups is One-way ANOVA or the Kruskal-Wallis test. We chose the One-way ANOVA. One-way analysis of variance (ANOVA) is the appropriate method instead of the t test, as the ANOVA is based on the same assumptions as the t test.

The only features that have a p-value >= 0.05 are histogram_number_of_zeroes and histogram_max. These features may have no predictive ability. All the other features may have predictive ability.

Lastly, we saved the cleaned data into a csv file for future use.

Preprocessing and Training Data Development

The following tasks were performed:

We started by importing the libraries needed for preprocessing. Downloaded the dataset ('archive/fetal_health.csv').

Reviewed the top five of the cleaned dataset, the dtypes, and summary statistics of the dataset.

The fhc_dataset has feature values that range from 0 - 1 or 0 - >100. The data needs to undergo min/max scaling to make it consistent and standardize the magnitude of numeric features using a scaler.

Scaling of the data using MinMaxScaler and performed a robust scaler fit_transform on the dataset in preparation to splitting the dataset.

We chose fetal_health as our dependent/response variable. Train test split() was applied to develop training data which will be used as our original training set in preparation for modeling. Target data was imbalanced, so we applied a stratified split based on fetal_health.

Train test split is a model validation procedure that allows you to simulate how a model would perform on new or unseen data. This splitting approach whether 70/30 or 80/20 split is the general rule for an effective holdout test data for model validation.

We used the 80/20 split in the model development dataset. We chose fetal_health as our dependent/response variable.

We checked the training and test set to ensure that the stratified split acted as expected.

Modeling

My Dataset and Applying the Machine Learning models:

The fetal health classification dataset is a multiclass classification problem. In multi-class classification, the goal is to classify the input into one of several classes or categories.

Classification problems that have multiple classes with imbalanced dataset present a different challenge than a binary classification problem. The skewed distribution makes many conventional machine learning algorithms less effective, especially in predicting minority class examples. To do so, let us first understand the problem at hand and then discuss the ways to overcome the imbalanced dataset.

The dataset is a multiclass classification. We are interested in RECALL measurement because it captures all instances of the positive class. The objective is to minimize false negatives and err on the side of caution. The ACCURACY, which is the proportion of the total number of correct predictions that were correct is being used as a tie breaker.

We would like a high positive conclusion even if it gives us a large number of false positives. We prefer a high false positive rate as opposed to high false negative rate, due to the problems of not being diagnosed as suspect or pathological.

Applying the Machine Learning models

Here are the following classification models we will be using:

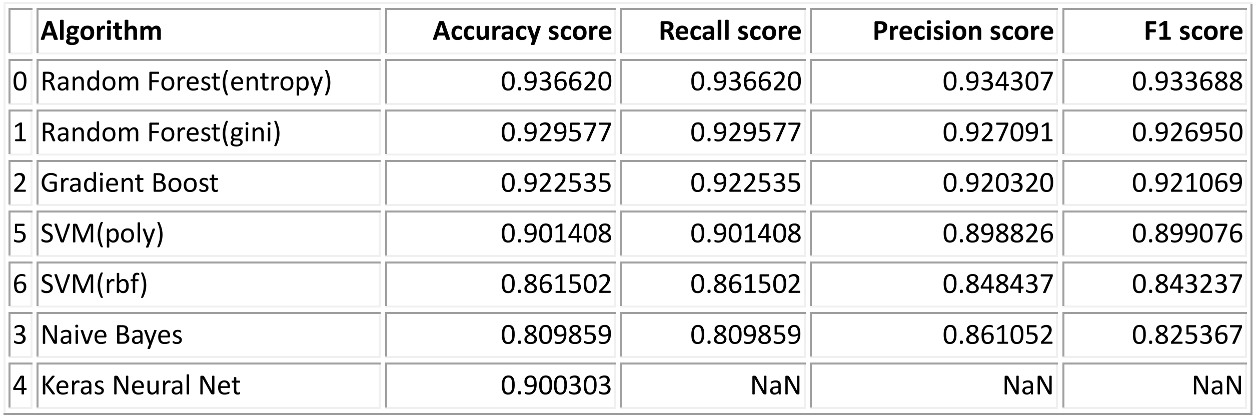
- Keras Neural Network
- Random Forest (Entropy)
- Random Forest (Gini)
- Gradient Boost
- Naive Bayes

- SVM (Polynomial)
- SVM (RBF)

Comparison of the models:

The top three algorithms based on Recall, Accuracy, Precision and F1 are:

- Random Forest (Entropy)
- Random Forest (Gini)
- Gradient Boost



	Algorithm	Accuracy score	Recall score	Precision score	F1 score
0	Random Forest(entropy)	0.936620	0.936620	0.934307	0.933688
1	Random Forest(gini)	0.929577	0.929577	0.927091	0.926950
2	Gradient Boost	0.922535	0.922535	0.920320	0.921069
5	SVM(poly)	0.901408	0.901408	0.898826	0.899076
6	SVM(rbf)	0.861502	0.861502	0.848437	0.843237
3	Naive Bayes	0.809859	0.809859	0.861052	0.825367
4	Keras Neural Net	0.900303	NaN	NaN	NaN

Algorithms Using the SMOTE Method

Imbalanced classification involves developing predictive models on classification datasets that have a severe class imbalance.

The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on, the minority class, although typically it is performance on the minority class that is most important.

One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples do not add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique or SMOTE for short.

We will apply the SMOTE method due to the target data being imbalanced.

Applying the SMOTE Method to the Classification Models:

The top three algorithms using SMOTE based on Recall, Accuracy, Precision and F1 are:

- Gradient Boost (SMOTE)
- Random Forest (Entropy)(SMOTE)
- Random Forest (Gini) (SMOTE)

	Algorithm	Accuracy score	Recall score	Precision score	F1 score
2	Gradient Boost(SMOTE)	0.934272	0.934272	0.932535	0.932919
0	Random Forest(entropy)(SMOTE)	0.931925	0.931925	0.931135	0.931400
1	Random Forest(gini)(SMOTE)	0.927230	0.927230	0.927471	0.927335
5	SVM(poly)(SMOTE)	0.901408	0.901408	0.920800	0.907318
6	SVM(rbf)(SMOTE)	0.852113	0.852113	0.901509	0.865727
3	Naive Bayes(SMOTE)	0.798122	0.798122	0.863455	0.818186
4	Keras Neural Net(SMOTE)	0.927299	NaN	NaN	NaN

Comparison of the Algorithms without and with SMOTE

The top three algorithms without and with the SMOTE Method based on Recall, Accuracy, Precision and F1 are:

- Random Forest (Entropy)
- Gradient Boost (Smote)
- Random Forest (Entropy)(SMOTE)

These models will be undergoing Hyperparameter Tuning.

	Algorithm	Accuracy score	Recall score	Precision score	F1 score
0	Random Forest(entropy)	0.936620	0.936620	0.934307	0.933688
2	Gradient Boost(SMOTE)	0.934272	0.934272	0.932535	0.932919
0	Random Forest(entropy)(SMOTE)	0.931925	0.931925	0.931135	0.931400
1	Random Forest(gini)	0.929577	0.929577	0.927091	0.926950
1	Random Forest(gini)(SMOTE)	0.927230	0.927230	0.927471	0.927335
2	Gradient Boost	0.922535	0.922535	0.920320	0.921069
5	SVM(poly)	0.901408	0.901408	0.898826	0.899076
5	SVM(poly)(SMOTE)	0.901408	0.901408	0.920800	0.907318
6	SVM(rbf)	0.861502	0.861502	0.848437	0.843237
6	SVM(rbf)(SMOTE)	0.852113	0.852113	0.901509	0.865727

	Algorithm	**Accuracy score**	**Recall score**	**Precision score**	**F1 score**
3	Naive Bayes	0.809859	0.809859	0.861052	0.825367
3	Naive Bayes(SMOTE)	0.798122	0.798122	0.863455	0.818186
4	Keras Neural Net	0.900303	NaN	NaN	NaN
4	Keras Neural Net(SMOTE)	0.927299	NaN	NaN	NaN

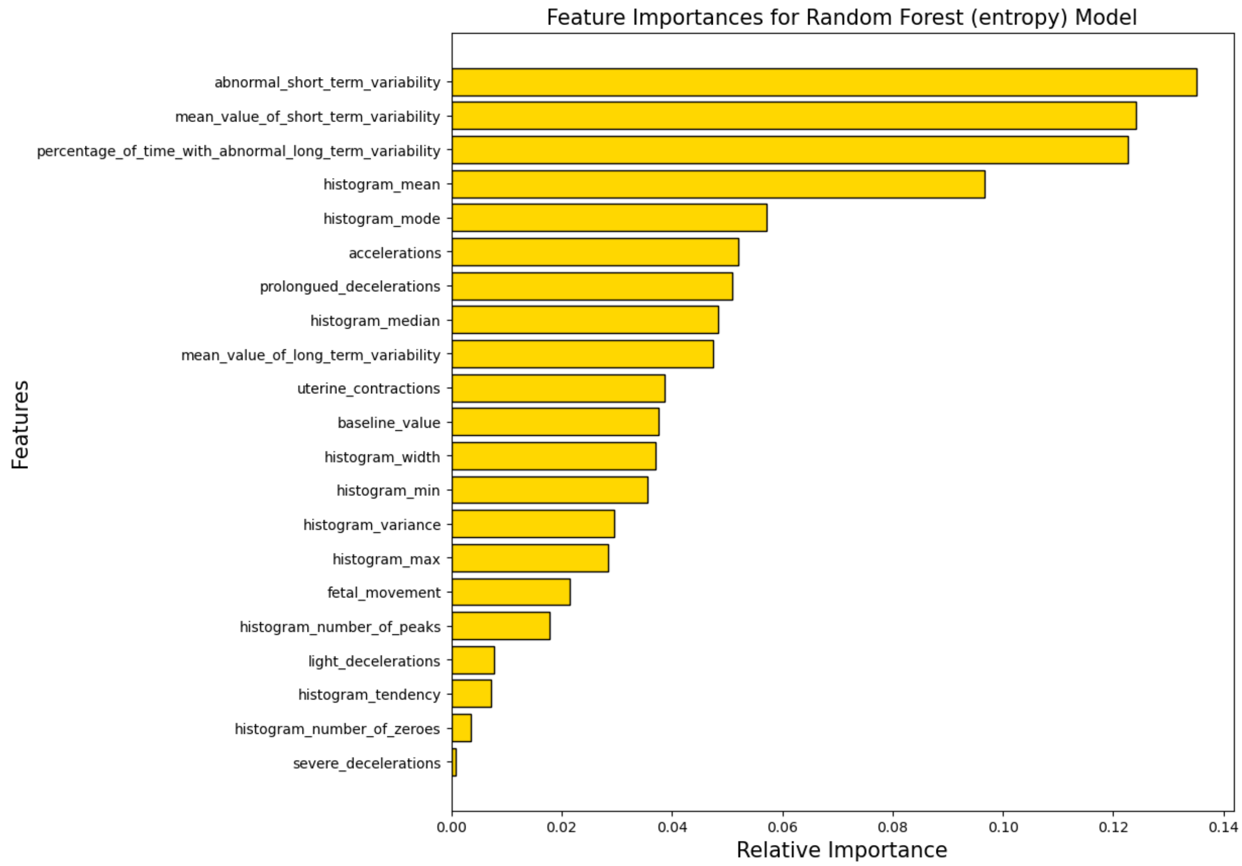
Top Three Algorithms to Undergo Hyperparameter Tuning

A. Hyperparameter Tuning of Random Forest (Entropy) Using RandomSearchCV

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits

RandomForestClassifier(n_estimators=1000, min_samples_split=2,
      min_samples_leaf=1, max_features= 'sqrt', max_depth=110,
      bootstrap=False)

Accuracy: 0.931924882629108
Recall: 0.931924882629108
Precision: 0.9302583757622245
F1: 0.9309027967974081
```

Next, we calculated the important features and found:

```
                                             Features  Importance scores
7                     abnormal_short_term_variability           0.135025
8                mean_value_of_short_term_variability           0.124042
9   percentage_of_time_with_abnormal_long_term_var...           0.122583
17                                     histogram_mean           0.096755
16                                     histogram_mode           0.057142
1                                       accelerations           0.052063
6                            prolongued_decelerations           0.050945
18                                   histogram_median           0.048443
10                mean_value_of_long_term_variability           0.047444
3                                 uterine_contractions           0.038738
0                                      baseline_value           0.037511
11                                    histogram_width           0.036993
12                                      histogram_min           0.035501
19                                 histogram_variance           0.029630
13                                      histogram_max           0.028532
2                                      fetal_movement           0.021579
14                          histogram_number_of_peaks           0.017765
4                                light_decelerations           0.007738
20                                 histogram_tendency           0.007150
15                          histogram_number_of_zeroes          0.003587
5                               severe_decelerations           0.000834
```



RESULTS: The top three features that predicts the fetal health classification are:

- abnormal_short_term_variability
- mean_value_of_short_term_variability
- percentage_of_time_with_abnormal_long_term_variability

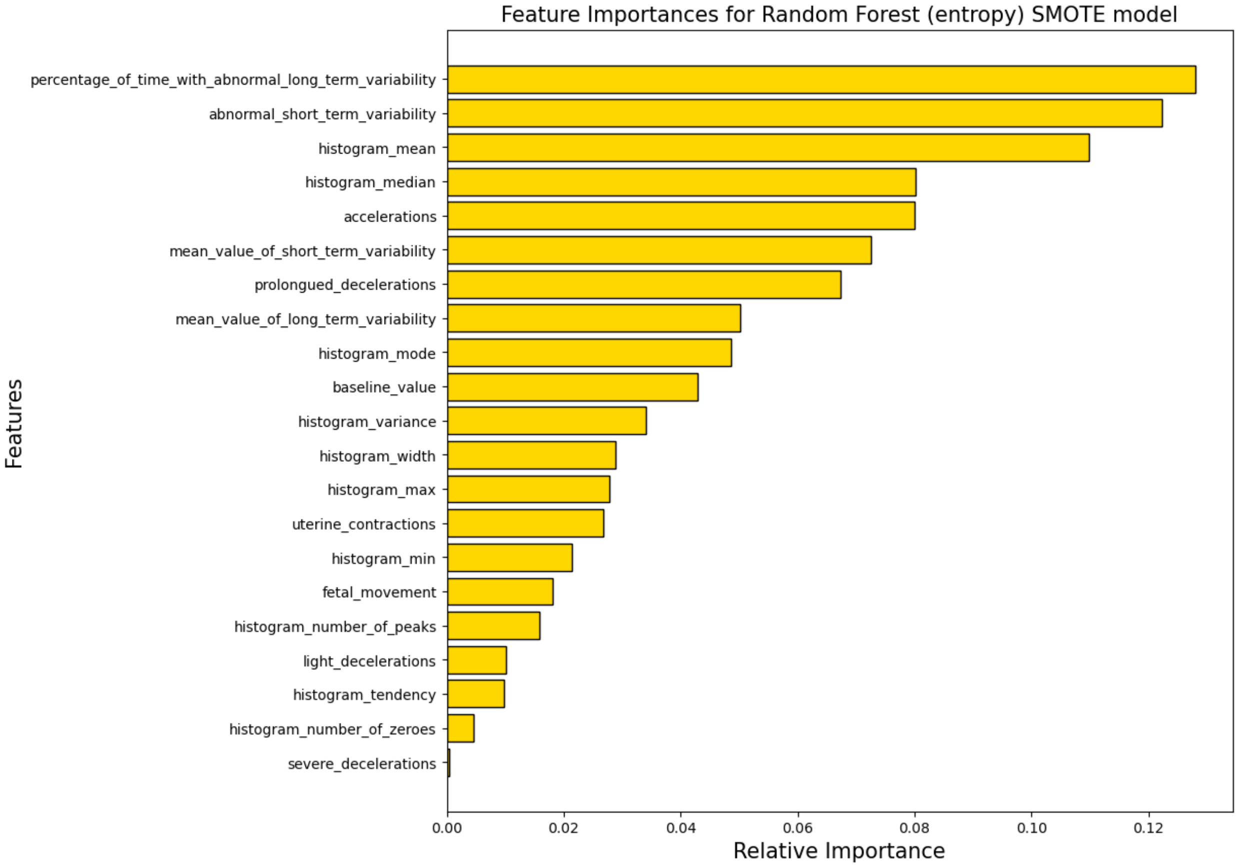
B Hyperparameter Tuning of Random Forest (Entropy) (SMOTE) Using RandomSearchCV

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits

RandomForestClassifier(n_estimators=1000, min_samples_split=2,
     min_samples_leaf=1, max_features= 'sqrt', max_depth=110,
     bootstrap=False)

Accuracy: 0.9295774647887324
Recall: 0.9295774647887324
Precision: 0.9287035023109206
F1: 0.9291024876260429
```

Next, we calculated the important features and found:

```
                                                Features  Importance scores
9   percentage_of_time_with_abnormal_long_term_var...           0.127952
7                        abnormal_short_term_variability           0.122235
17                                        histogram_mean           0.109764
18                                      histogram_median           0.080230
```

```
1                           accelerations           0.080064
8       mean_value_of_short_term_variability           0.072576
6                   prolongued_decelerations           0.067370
10       mean_value_of_long_term_variability           0.050223
16                            histogram_mode           0.048615
0                             baseline_value           0.042910
19                        histogram_variance           0.034075
11                           histogram_width           0.028859
13                             histogram_max           0.027770
3                        uterine_contractions           0.026792
12                             histogram_min           0.021460
2                             fetal_movement           0.018121
14                 histogram_number_of_peaks           0.015931
4                        light_decelerations           0.010186
20                        histogram_tendency           0.009827
15                histogram_number_of_zeroes           0.004583
5                       severe_decelerations           0.000458
```



RESULTS: The top three features that predicts the fetal health classification are:

- percentage_of_time_with_abnormal_long_term_variability
- abnormal_short_term_variability
- histogram_mean

C. Hyperparameter Tuning of Gradient Boost (SMOTE) Using RandomGridSearchCV

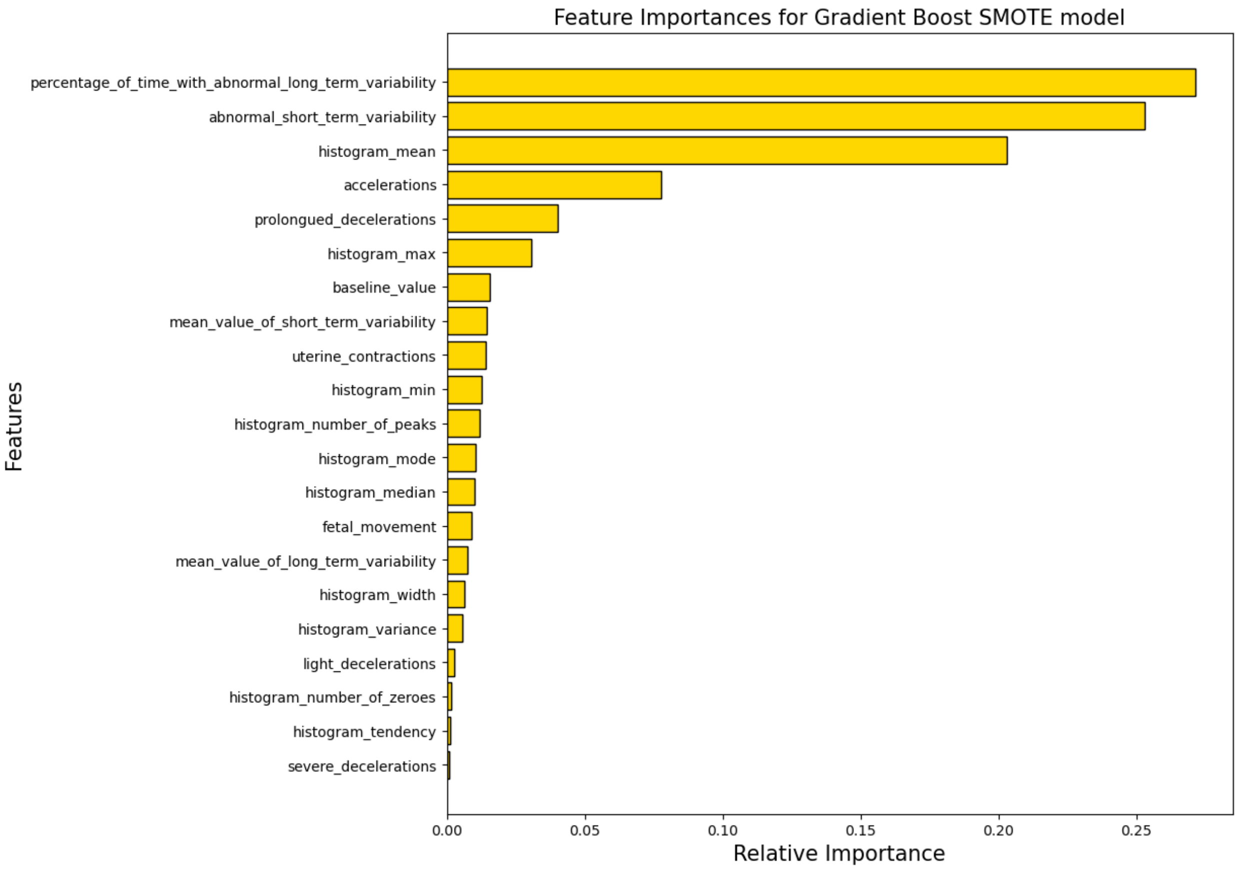
```
Fitting 3 folds for each of 81 candidates, totalling 243 fits
```

```
GradientBoostingClassifier(learning_rate=0.05, max_depth=8,
      max_leaf_nodes=110, n_estimators=180, random_state=5,
      subsample=0.8)
```

```
0.9790879314688837
```

Next, we calculated the important features and found:

```
                                             Features  Importance scores
9   percentage_of_time_with_abnormal_long_term_var...           0.271332
7                     abnormal_short_term_variability           0.253091
17                                     histogram_mean           0.202921
1                                       accelerations           0.077726
6                            prolongued_decelerations           0.040268
13                                      histogram_max           0.030624
0                                      baseline_value           0.015532
8                mean_value_of_short_term_variability           0.014398
3                                uterine_contractions           0.014152
12                                      histogram_min           0.012492
14                          histogram_number_of_peaks           0.011875
16                                     histogram_mode           0.010502
18                                   histogram_median           0.009924
2                                      fetal_movement           0.008975
10                mean_value_of_long_term_variability           0.007465
11                                    histogram_width           0.006279
19                                 histogram_variance           0.005641
4                                 light_decelerations           0.002902
15                         histogram_number_of_zeroes           0.001784
20                                 histogram_tendency           0.001154
5                                severe_decelerations           0.000963
```



RESULTS: The top three features that predicts the fetal health classification are:

- percentage_of_time_with_abnormal_long_term_variability
- abnormal_short_term_variability
- histogram_mean

Comparison of the three algorithms that underwent Hyperparameter Tuning:

	Algorithm	Accuracy score	Recall score	Precision score	F1 score
2	Gradient Boost(SMOTE)	0.931925	0.931925	0.930814	0.931189
0	Random Forest(entropy)	0.931925	0.931925	0.930258	0.930903
1	Random Forest(entropy)(SMOTE)	0.929577	0.929577	0.928704	0.929102

After hyperparameter tuning the top three algorithms, **Gradient Boost (SMOTE)** and **Random Forest (Entropy)** showed the best results which is as follows:

Both algorithms have the same Recall and Accuracy Score:

- Accuracy: 0.9319
- Recall: 0.9319

Based on the Precision and F1 Score the **Gradient Boost (SMOTE)** is the best model:

- Precision: 0.9308
- F1: 0.9311

The top three features of importance that predicts the fetal health classification based on **Gradient Boost (SMOTE)** are:

- percentage_of_time_with_abnormal_long_term_variability
- abnormal_short_term_variability
- histogram_mean

The top three models showed that whether the model underwent SMOTE or not, the results were still fairly close in accuracy, recall, Precision and F1. Despite the different ranking, the features of importance had almost the same results.

Summary

After the data wrangling, EDA and preprocessing of the fetal health classification dataset, we identified this as an unbalanced Multiclass Classification.

A Multiclass Classification is a classification with more than two classes, and each sample can only be labeled as one class. In multiclass classification, each record belongs to one of three or more classes, and the algorithm's goal is to construct a function which, given a new data point, will correctly identify the class into which the new data point falls. To handle this particular problem, we can use a machine learning algorithm for multiclass classification like Neural Networks, Naive Bayes, and SVM.

Imbalanced dataset in a multiclass classification means, if one class has overwhelmingly more samples than another, it can be seen as an imbalanced dataset. That is, a classification problem where the classes are not represented equally. This imbalance causes two problems: Training is inefficient as most samples are examples that contribute no useful learning signal. The examples can overwhelm training and lead to degenerate models.

We have chosen to use the SMOTE method to handle the class imbalance. Oversampling is one of the most widely used techniques to deal with imbalance classes. In SMOTE, we generate synthetic samples for minority classes to make sure we have enough data to train the model.

First, we chose seven different models to apply the fit method. We chose Keras Neural Network Model, Random Forest (entropy), Random Forest (Gini), Gradient Boost, Naïve Bayes, SVM (Polynomial) and SVM (RBF).

After training the data using the different algorithms and applying them to the test data, the results are as follows:

The top three algorithms without SMOTE based on Recall:

- Random Forest (Entropy) Recall:0.9366
- Random Forest (Gini) Recall:0.9295
- Gradient Boost Recall:0.9225

The top three algorithms using SMOTE based on Recall are:

- Gradient Boost (SMOTE) Recall: 0.9342
- Random Forest (Entropy)(SMOTE) Recall:0.9319
- Random Forest (Gini) (SMOTE) Recall: 0.9272

The top three algorithms without and with using the SMOTE Method based on Recall are:

- Random Forest (Entropy) Recall: 0.9366
- Gradient Boost (SMOTE) Recall: 0.9342
- Random Forest (Entropy)(SMOTE) Recall: 0.9319

In order to understand these algorithms, let us talk about Random Forest and Gradient Boost.

Random Forest is a type of ensemble learning method for classification, regression, and other tasks. Random Forests works by constructing many decision trees at training time. The way that this works is by averaging several decision trees at different parts of the same training set.

Entropy is an information theory metric that measures the impurity or uncertainty in a group of observations. It determines how a decision tree chooses to split data. In the context of Decision Trees, entropy is a measure of disorder or impurity in a node. Thus, a node with more variable composition, such as 2Pass and 2 Fail would be considered to have higher Entropy than a node which has only pass or only fail.

Gradient boosting is one of the variants of ensemble methods where you create multiple weak models and combine them to get better performance as a whole.

There are two main differences between the gradient boosting trees and the random forests. We train the former sequentially, one tree at a time, each to correct the errors of the previous ones. In contrast, we construct the trees in a random forest independently.

These models will be undergoing Hyperparameter Tuning.

After hyperparameter tuning the top three algorithms, **Gradient Boost (SMOTE)** and **Random Forest (Entropy)** showed the best results which is as follows:

Both algorithms have the same Recall and Accuracy Score:

- Accuracy: 0.9319
- Recall: 0.9319

Based on the Precision and F1 Score the **Gradient Boost (SMOTE)** is the best model:

- Precision: 0.9308
- F1: 0.9311

Comparison of the EDA's correlation of features with fetal health and the actual outcome from the top model are:

From the EDA, the correlation of the different features against fetal_health and the top three that have the strongest correlation with fetal_health are:

- prolongued_decelerations
- abnormal_short_term_variability
- percentage_of_time_with_abnormal_long_term_variability

After modeling and hyperparameter tuning, the outcomes in the test dataset are as follows:

The top three feature of importance that predicts the fetal health classification based on **Gradient Boost (SMOTE)** are:

- percentage_of_time_with_abnormal_long_term_variability
- abnormal_short_term_variability
- histogram_mean

Gradient Boost (SMOTE) is the best of all the algorithms performed that would accurately predict fetal health outcomes accurately using Cardiotocograms (CTGs) data to prevent or decrease child and maternal mortality.

Recommendations to Client based on the Results:

The goal is to predict fetal outcomes using CTGs in order to prevent or decrease child and maternal mortality.

The features measured many trends in fetal heart rate such as variations, accelerations and decelerations and histogram values of the heart rates. As well as uterine contractions and fetal movements.

In our best model Gradient Boost (SMOTE), the features of importance are:

- Percentage of time with abnormal long-term variation, which is a similar measurement to short term variability except for the duration of time.
- Abnormal short-term variability or STV is a measurement of the beat-to-beat variation in the fetal heart rate, and the CTG keeps track of the number of times the variation is abnormal in a fetus.
- The histogram values represent the baby's heart rates that were measured, all plotted on a histogram.
- CTGs are cost-effective and highly accessible around the world, even in places where healthcare might be harder to access.

Using the Gradient Boost (SMOTE) model, the CTG can be easily inputted and classified, which can help reduce child mortality. On top of that, many places around the world can obtain access to these machines without having to worry about finding an expert obstetrician who can read the measurements from the machine learning algorithm. A simple, easy to use model such as the one created will encourage the use of more CTGs, thereby further preventing child mortality and allowing early diagnoses of many different fetal conditions.

Using machine learning models, fetal health classifications can be made without the presence of obstetricians and in a much timelier manner. These models can be highly accurate in their predictions, making them viable solutions to the problem of fetal health, while this solution works theoretically, there are still some major issues with implementing machine learning models.

Why this Model is Highly Recommended:

This model would be able to accurately predict which measurements from the cardiotocogram were the most influential in the predictions for feature importance.

This model will certainly be beneficial for clinical use and will allow healthcare providers to specify their diagnosis and treatment specifically towards the detected problem, especially for suspect and pathological categories.

Future Investigations

The models showed wonderful results but there is always room to grow:

- Feature engineering using PCA
- Other Cross Validation techniques like Stratified K-FoldCV method, Combination of Random SearchCV and Grid SearchCV method
- Techniques to check outliers and reduce possible bias for the dataset
- Using Keras Neural Networks methods with hyperparameter tuning using more epochs, different batch sizes and optimizers
- Using SHapley Additive exPlanations (SHAP), Local Interpretable Model Agnostic Explanations (LIME) and Feature Altering for explanations of Black box models (FAB) are explainability methods to increase interpretability of the results returned by the machine learning model and to determine which features from the data are most influential when making a prediction and this will lead to both patients and medical professionals to trust the classifications made by the model
- Investigation of other techniques or methods by taking into consideration computational complexity, cost effectiveness and maintenance cost