Prediction of Fetal Health Status from Cardiotocography

Austin Wong

Western Governors University

C964: Computer Science Capstone

February 2021

Contents

Tables

Figures

Section A

Letter of Transmittal

February 17, 2021

Dr. Onome Ako
Director, Amref Health Africa
2100 W 129th Ave
Seattle, WA 98113

Dear Dr. Ako,

Decreasing infant mortality rates continue to be a challenge for developing countries. Fetal distress is one significant factor of infant mortality that can be addressed by improved fetal health monitoring. While cardiotocographs (CTGs) serve as an important screening tool to assess fetal health, the correct interpretation of CTGs continues to be a problem for obstetricians.

With the attached proposal, you will find a solution that automates the interpretation of CTG data so that obstetric care teams can spend less time deciphering CTGs and more time focusing on the delivery of high-quality care in stressful environments. By automatically monitoring fetal health, critical conditions can be detected and resolved earlier leading to fewer perinatal deaths.

The proposed solution provides an application that places the fetus's health status into one of three categories: Normal, Suspect, or Pathologic. When fetal health is classified as Suspect or Pathologic, care teams can evaluate which interventions to take to ensure safe delivery. The solution will aim to predict fetal health status with an accuracy of at least 85%.

The total cost of the proposed solution is $5,720. This budget includes requirements planning, application development, testing, implementation, and deployment for 104 hours.

Thank you for your consideration of this solution. We appreciate your business and look forward to working with you. Please review the proposal and respond with your thoughts.

If you have any questions, please feel free to contact me at austin.wong@wongsolutions.com or (303) 825-1664. I look forward to speaking with you about this project in further detail.

Sincerely,

Austin Wong

Austin Wong

Project Recommendation

Problem Summary

One of the leading causes of death in children under 5 years is birth asphyxia/trauma. Effective monitoring of fetal health can lead to quicker response times for life-saving interventions in these scenarios. Cardiotocographs (CTGs) act as an important tool to assess and monitor fetal health leading up to and during childbirth (Chandraharan & Arulkumaran, 2007). However, the interpretation of CTGs continues to vary among obstetricians, creating a need for a more objective method of CTG interpretation (Murphy et al., 1990).

The proposed solution will provide an application that automates CTG interpretation, placing the fetus's health status into one of three categories: Normal (no intervention is needed), Suspect (intervention may be needed), or Pathologic (intervention is vital). A baseline model will be included as well as the capability to create new models based on new data.

Application Benefits

The application will provide an automated interpretation of CTGs that produces a more consistent determination of fetal health status and frees up the obstetric care team to handle other important tasks. By correctly identifying "Suspect" and "Pathologic" fetal health statuses more frequently, care teams will be able to intervene to prevent more deaths in children under 5 years, thereby decreasing infant mortality rates.

Application Description

The application will take 21 CTG data points as input and use a machine learning model to output a prediction of fetal health status (Normal, Suspect, or Pathologic). Users will be able to access visualizations that compare the user-entered CTG attributes to the data used to build the model. The application will be able to add new CTG data to the database, which can then be used to create updated models.

Data Description

The data that will be used to create the application's machine learning model originated from a study published in The Journal of Maternal-Fetal Medicine (Ayres-de-Campos, 2000). To create the dataset, 2126 CTGs were processed into 21 numeric attributes of fetal heart rate, fetal movement, and uterine contractions. Each CTG was also classified as one of three fetal health states (Normal, Suspect, Pathologic) by three expert obstetricians. All this information was supplied in the form of a CSV file.

The 21 numeric attributes will be used as independent variables to create a machine learning model that classifies fetal health state – the dependent variable. Upon inspection of the data, no anomalies in the form of missing values, erroneous values, or outliers were found. Approximately 78%, 14%, and 8% of the CTGs were classified as Normal, Suspect, and Pathologic, respectively. As a result, there will be fewer Suspect and Pathologic cases to train and test the machine learning model. The dataset does not take into consideration risk factors for high-risk pregnancies such as maternal age or pre-existing conditions.

Objective and Hypotheses

Objectives of the application include the application acceptance of CTG data input, the prediction of fetal health status, and the ability to update the model via the addition of new data. These objectives will be pursued with the goals of increasing detection rates of abnormal fetal health statuses during birth and expediting detection time of abnormal fetal health statuses. I propose that if obstetric care teams use the application's automated fetal health status prediction during childbirths, then the number of complications during labor that these care teams experience will decrease, thereby reducing the infant mortality rate. I also hypothesize that if the model is trained with 1,700 CTGs and tested with 426 CTGs, then it will be able to achieve a macro F1-score of at least 85%.

Methodology

The waterfall development methodology will be used to manage this project. While other methodologies may produce a usable product in a shorter amount of time, this methodology will lead to a complete product when it is first used. This will be important because of the nature of the field of obstetrics where errors can have large consequences on human lives. Furthermore, the waterfall method will be a good approach because of the small size of the project.

The requirements gathering and analysis phase will correspond with initial discussions regarding the required features of the project. These discussions will involve input from obstetric care teams, particularly expert obstetricians. The system design phase will correspond with the determination of the application architecture and data models. During the implementation phase, the actual coding of the application and configuration of the database will take place. The application will be tested and debugged in the testing phase before being delivered and installed on Amref's partnered hospitals' workstations in the deployment phase. Following deployment, the project will enter the maintenance phase where new issues are fixed. The project will proceed in the order above with each phase completing before the beginning of the next phase.

Funding Requirements

The project will have no hardware costs because the application will operate on the hospitals' existing workstations. The total cost of software engineering resources will be $5,720 for 104 hours of work at $55 per hour. The project will not have any costs associated with licensing fees because the development tools and software libraries utilized will be free to use. Therefore, the total project cost will be $5,720, excluding annual maintenance costs of $1,100 per year.

Stakeholders Impact

This application will allow obstetric care teams to spend less time interpreting CTGs and more time focusing on delivering high-quality care. Furthermore, this application will lead to less variability in CTG interpretation at the organizational level for hospitals, which is expected to lead to more consistent and appropriate decision-making during childbirth. On an individual level, this means that fetuses will be more likely to survive and will be less likely to experience negative health outcomes, which also benefits the mother and other family members. On a population level, this is expected to help lower infant mortality rates. These results could lead to a higher likelihood of Amref receiving future grants and funding.

Data Precautions

The data used in the application will not include any identifying attributes of patients. Therefore, the data will not be considered sensitive nor PHI. As such, HIPAA compliance will not apply to this project.

Developer’s Expertise

The software developer completing this project has 10 years of experience in engineering software solutions for healthcare organizations. The developer has a Bachelor of Science degree in Computer Science and a Master of Science degree in Health Information Management. Furthermore, the developer has professional certificates from Stanford University and Google in Machine Learning and Machine Learning with TensorFlow on Google Cloud Platforms, respectively.

Section B

Project Proposal

Problem Statement

Variability and inaccuracy of cardiotocograph (CTG) interpretation is a problem faced by obstetric care teams (Murphy et al., 1990). Consistent and accurate analysis of CTGs is needed to determine fetal health status to identify and resolve childbirth complications such as birth asphyxia (Chandraharan & Arulkumaran, 2007). Benefits of an automated analysis system include improved healthcare delivery in the obstetric environment and lower infant mortality rate as a result of more accurate and earlier identification of fetal distress.

Customer Summary

Users of this desktop application will be obstetricians as well as other physicians and nurses on the obstetric care team. This application will be used on workstations in both triage and delivery rooms in Amref's partner hospitals. At a minimum, users will need basic computer navigation skills and an understanding of the attributes of a CTG to input the attributes for the application to perform its analysis and determine fetal health status. The care team will benefit from this automated analysis by conserving their focus for other critical tasks in the obstetric environment. Furthermore, the automated analysis will provide a more consistent categorization of fetal health status than human analysis, which will lead to earlier detection of fetal distress and allow care teams to intervene at an earlier time.

Existing System Analysis

Currently, Amref's partner hospitals use workstations connected to electronic fetal monitoring equipment that provides a readout of CTG attributes. This system allows the care team to review and interpret CTGs themselves but does not interpret the graphs for them. The workstations use Windows 10 as their operating system.

Upon completion of the project, the proposed application and its accompanying database will be installed on these workstations. Furthermore, SQLite 3.34.1 and Python 3.8.5 will be installed with the following libraries: Joblib 1.0.0, Matplotlib 3.3.2, NumPy 1.19.2, Pandas 1.2.1, PySimpleGUI 4.34.0, Scikit-learn 0.23.2, Seaborn 0.11.1, and SQLite 3.33.0**.**

Data

The data used in the application's machine learning model will come from Kaggle (Maranhão, 2020). However, the dataset originated from The Journal of Maternal-Fetal Medicine (Ayres-de-Campos et al., 2000) and was uploaded to the UCI Machine Learning Repository (Dua & Graff, 2019). To create the dataset, 2126 CTGs were processed into 21 numeric attributes of fetal heart rate, fetal movement, and uterine contractions. Each CTG was also classified as one of three fetal health states (Normal, Suspect, Pathologic) by three expert obstetricians. All this information was supplied in the form of a CSV file.

The 21 attributes will be used as independent variables to create a machine learning model that classifies fetal health state – the dependent variable. The data will be analyzed for null values, erroneous values, and outliers. Rows that fall into any of these three categories will be removed from the dataset. This dataset will be used to create a baseline machine learning model used by the application. Data that will be added by users to create new machine learning models will undergo input validation before being added to the database.

Project Methodology

This project will be managed using the traditional waterfall development methodology. This method was chosen because it is cost-effective for small projects and requirements are not expected to change once established. Furthermore, the users will need a complete working product, not a smaller deliverable, before they use it in the delivery room.

First, requirements gathering will involve conversations about the needs of obstetric care teams as it relates to how they interact with the application and what kind of outputs will be useful to them. Then, the project will proceed with the design phase where the application architecture will be determined and the data models defined. Next, coding of the application, the configuration of the database, and unit testing will take place during the implementation phase. The remaining testing and debugging will occur during the testing phase. This will include integration and acceptance testing. Once the application passes the testing phase, it will be delivered and installed on Amref's workstations in the deployment phase. Following deployment, the project will enter the maintenance phase where issues not previously discovered in the testing phase will be resolved. The project will proceed in the order above with each phase completing before the beginning of the next phase.

Project Outcomes

Product deliverables will include a desktop application with a graphical user interface that allows users to input CTG data, predict fetal health states, save new data to the database, access a dashboard to analyze and visualize the CTG data, and train new models that incorporate recently added data. A database that houses CTG data and user login data will be included as well. A user guide will be provided to explain the user interface as well as provide installation instructions. A project schedule detailed with milestones, estimated completion dates, and actual completion dates will be included as a product deliverable.

Implementation Plan

The software application will be implemented using the existing hardware in Amref's partnered hospitals. The application, database, and their dependencies will be installed onto the workstations used in triage and delivery rooms. The software will be rolled out all at once so that all teams can be trained on the new software and integrate it into their practices at the same time. User accounts to log in to the application will be created in the database before training.

Except unit testing, all testing will occur during the testing phase. This will include user acceptance testing in a simulation of the actual delivery room environment. Because this project will be using the waterfall method, it will be important that end users are involved during the requirements gathering phase. All major issues discovered during this phase will be resolved before the final distribution.

Dependencies and milestones will be covered in the Timeline and Milestones section on page ***. After implementation, the obstetric care teams in Amref's partner hospitals will have the ability to predict fetal health status using CTG data via an automated analysis desktop application. Furthermore, they will be able to train new models to stay updated with new data, and they will be able to analyze trends in the data using the GUI's dashboard.

Evaluation Plan

The application will undergo unit testing and integration testing for verification and acceptance testing as part of validation. In addition to testing, the application will be evaluated based on its macroaveraged F1-score for its machine learning model. This metric will be used instead of accuracy because while fetal distress is a problem that needs to be addressed, it is an uncommon complication of labor, meaning the dataset will likely be imbalanced. Furthermore, predicting pathologic fetal health statuses precisely and with high recall will have greater value to care teams than accurately predicting normal fetal health statuses. A macro F1-score of 85% or higher will indicate project success.

Resources and Costs

Hardware*:

- Workstations
  - Windows 10 OS
  - Minimum 8.0 GB RAM

Software**:

- PyCharm IDE
- SQLite 3.34.1
- Python 3.8.5 and the following libraries:
  - Joblib 1.0.0
  - Matplotlib 3.3.2
  - NumPy 1.19.2
  - Pandas 1.2.1
  - PySimpleGUI 4.34.0
  - Python 3.8.5
  - Scikit-learn 0.23.2
  - Seaborn 0.11.1
  - SQLite 3.33.0

Human resources:

- 1 Software Engineer

*Hardware will be provided by Amref and is not included in cost estimates

**The libraries and applications required for this project are free to use without additional licensing fees

Table 1: Itemized Breakdown of Resources and Costs

Description	Rate	Time	Subtotals
Hardware	---	---	---
Software	---	---	---
Human Resource: Software Engineer – Development and Testing of the product	$55 /hour	104 hours	$5,720
Human Resource: Software Engineer – Maintenance	$1,100/year	Maximum 40 hours/year, 1-year minimum	$1,100
		Total	**$6,820**

Timeline and Milestones

Table 2: Project Timeline and Milestones

#	Dependencies	Milestone	Resources	Hours	Start	End
1	None	Requirements completed	Software Engineer, Stakeholders	15	3/1	3/2
2	1	Architecture Design	Software Engineer	10	3/3	3/4
3	1	Database Design	Software Engineer	10	3/5	3/8
4	3	Database Creation	Software Engineer	3	3/9	3/9
5	4	Data Analysis Complete	Software Engineer	10	3/10	3/11
6	2, 5	Code Development Complete	Software Engineer	25	3/12	3/17
7	6	Testing Complete	Software Engineer, End Users	15	3/18	3/22
8	7	Application Deployment	Software Engineer, Stakeholders	8	3/23	3/24
9	8	Final Project Delivery	Software Engineer, Stakeholders	8	3/25	3/26
			Total	**104**	**3/1**	**3/26**

Section D

Post-Implementation Report

Project Purpose

The purpose of this project was to reduce infant mortality rates in Amref's partner hospitals by automating the analysis of fetal health attributes provided by cardiotocography (CTG). Amref wanted to focus their efforts on fetal distress in the moments before and during childbirth. To achieve this goal, an application that uses machine learning to predict the category of fetal health status (FHS) was designed, developed, tested, and implemented.

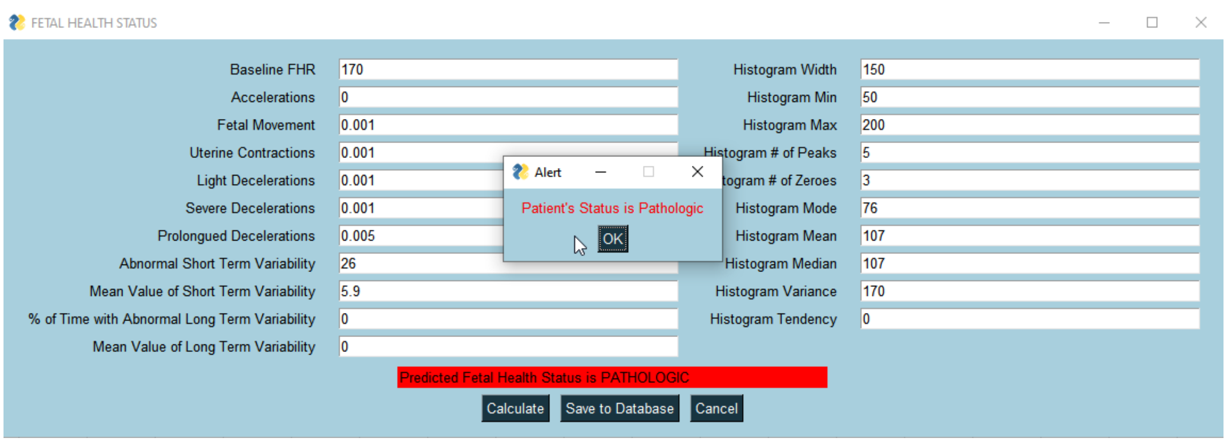

Figure 1: Fetal Health Status Prediction from CTG Input

Using this application, obstetric care teams were able to analyze the fetal dataset to better understand various patterns in the CTG data with relation to fetal health status. They were provided with a user-friendly dashboard that allowed them to look at trends in baseline fetal heart rate (FHR), accelerations, and prolonged decelerations in the overall population as well as by fetal health status. Additionally, users could add new patient data to the database and train a new model that includes the newly updated dataset. As a result of this project, care teams were able to more quickly identify patients classified as suspect and pathologic, enabling them to intervene earlier to ensure safe delivery.

Datasets

The dataset used in this project came from Kaggle. The raw dataset from *** insert paper reference *** had minor differences from the Kaggle dataset. The primary differences were different column names and the removal of extraneous information in the Kaggle dataset. Exploration of the data using descriptive statistics and scripts to search for null values, erroneous values, and outliers led to the conclusion that cleaning and manipulation of the data were not necessary. There were 2126 rows each representing a patient with complete CTG data. The "fetal_health" and "histogram_tendency" columns contained categorical data that took on numeric values while the remaining 20 columns contained numerical data.

baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	mean_value_of_short_term_variability	percentage_of_time_with_abnormal_long_term_variability
120	0	0	0	0	0	0	73	0.5	43
132	0.006	0	0.006	0.003	0	0	17	2.1	0
133	0.003	0	0.008	0.003	0	0	16	2.1	0
134	0.003	0	0.008	0.003	0	0	16	2.4	0
132	0.007	0	0.008	0	0	0	16	2.4	0
134	0.001	0	0.01	0.009	0	0.002	26	5.9	0
134	0.001	0	0.013	0.008	0	0.003	29	6.3	0
122	0	0	0	0	0	0	83	0.5	6
122	0	0	0.002	0	0	0	84	0.5	5
122	0	0	0.003	0	0	0	86	0.3	6

Figure 2: Cropped Screenshot of Kaggle Dataset

FileName	Date	SegFile	b	e	LBE	LB	AC	FM	UC	ASTV	MSTV	ALTV	MLTV	DL	DS	DP	DR
Variab10.txt	12/1/1996	CTG0001.txt	240	357	120	120	0	0	0	73	0.5	43	2.4	0	0	0	0
Fmcs_1.txt	5/3/1996	CTG0002.txt	5	632	132	132	4	0	4	17	2.1	0	10.4	2	0	0	0
Fmcs_1.txt	5/3/1996	CTG0003.txt	177	779	133	133	2	0	5	16	2.1	0	13.4	2	0	0	0
Fmcs_1.txt	5/3/1996	CTG0004.txt	411	1192	134	134	2	0	6	16	2.4	0	23	2	0	0	0
Fmcs_1.txt	5/3/1996	CTG0005.txt	533	1147	132	132	4	0	5	16	2.4	0	19.9	0	0	0	0
Fmcs_2.txt	5/3/1996	CTG0006.txt	0	953	134	134	1	0	10	26	5.9	0	0	9	0	2	0
Fmcs_2.txt	5/3/1996	CTG0007.txt	240	953	134	134	1	0	9	29	6.3	0	0	6	0	2	0
Hasc_1.txt	2/22/1995	CTG0008.txt	62	679	122	122	0	0	0	83	0.5	6	15.6	0	0	0	0
Hasc_1.txt	2/22/1995	CTG0009.txt	120	779	122	122	0	0	1	84	0.5	5	13.6	0	0	0	0
Hasc_1.txt	2/22/1995	CTG0010.txt	181	1192	122	122	0	0	3	86	0.3	6	10.6	0	0	0	0

Figure 3: Cropped Screenshot of Raw Dataset

Data Product Code

The data was initially explored and analyzed in a Jupyter Notebook. The Pandas library was used to load the data into a Pandas DataFrame. The Matplotlib library was used to produce initial visualizations. The Scikit-Learn library was used to create machine learning models. A correlation matrix was created to identify the strongest associations between the features and the "fetal_health" label in the dataset. Various histograms were created to visualize these associations with all the data as well as with subsets of the data split by fetal health status. The code used to create these visualizations was used in the dashboard component of the application.

The data was split into training and test data (80% / 20%, respectively) to use for training and evaluation of machine learning models that predict fetal health status. Random forest classifiers were explored using the original data as well as using normalized and standardized data. A classification report including precision, recall, f1-score, accuracy, and support for each fetal health status, macro averages, and weighted averages was generated. Comparing the reports led to the conclusion that differences in model performance amongst original, normalized, and standardized data were negligible, so original data was used for the remainder of the project.

Other explored classifiers included Support Vector Machine (SVM), Naïve Bayes, Stochastic gradient descent (SGD), Logistic Regression, and k-nearest neighbors (KNN). Random forest was the best performing algorithm for this dataset. The RandomizedSearchCV and GridSearchCV classes from Scikit-Learn were used to tune hyperparameters for the random forest classifier. Because the imbalanced dataset provided a majority of normal fetal health statuses and few pathologic fetal health statuses, Macro F1-scores were used to evaluate predictions on the test set with 5-fold cross-validation. Based on these results, the initial untuned random forest classifier performed best. This model was incorporated into the final product.

However, users were able to train new models that reflect the latest data using the application. New data submitted to the database was subjected to input validation before incorporating it into the dataset, eliminating the need for preprocessing later. The application code followed the same process of splitting the data, training tuned and untuned random forest classifiers using 5-fold cross-validation, and comparing macro F1-scores to produce a new model.

Hypothesis Verification

Two hypotheses were presented in this project:

1. If the model is integrated into the workflow of obstetric care teams in Amref's partner hospitals, then it will lead to a decrease in the infant mortality rate.
2. The model will be able to classify fetal health status with a macro F1-score of at least 85% if trained with 1,700 CTGs and tested with 426 CTGs.

The first hypothesis cannot be verified yet. Further studies that evaluate the infant mortality rate over an extended period and control for other contributing factors will be necessary before accepting or rejecting this hypothesis.

The second hypothesis was accepted. The model received a macro F1-score of 91% on both training and test sets, which divided the 2,126 CTGs into 1,700 and 426 CTGs, respectively.

Effective Visualizations and Reporting

The initial dataset in a CSV file was directly loaded into an SQLite database table. New patient data added by application users went through input validation to ensure the data had appropriate values for each attribute before inserting it into the database.

A bar chart was used to illustrate the relative distribution of the data in each fetal health category. Visualizations were consistently color-coded to depict each category. Green, yellow, and red were used to represent normal, suspect, and pathologic fetal health statuses, correspondingly.

As discussed in the Data Product Code section, a correlation heatmap was used to determine which components of the CTG were most strongly correlated with fetal health status. Cells moved from dark blue to green to light yellow as their correlations moved from strong positive correlations to weak correlations to strong negative correlations. Based on this analysis, histograms were created to compare trends in some of the most strongly correlated components (Baseline FHR, Accelerations, and Prolonged Decelerations) using all the data as well as subsets of the data split by fetal health status. The y-axis scale varied by the graph to illustrate trends and distributions of data within each fetal health status subset. Otherwise, with a fixed scale the data representing normal fetal health status would dominate the space.

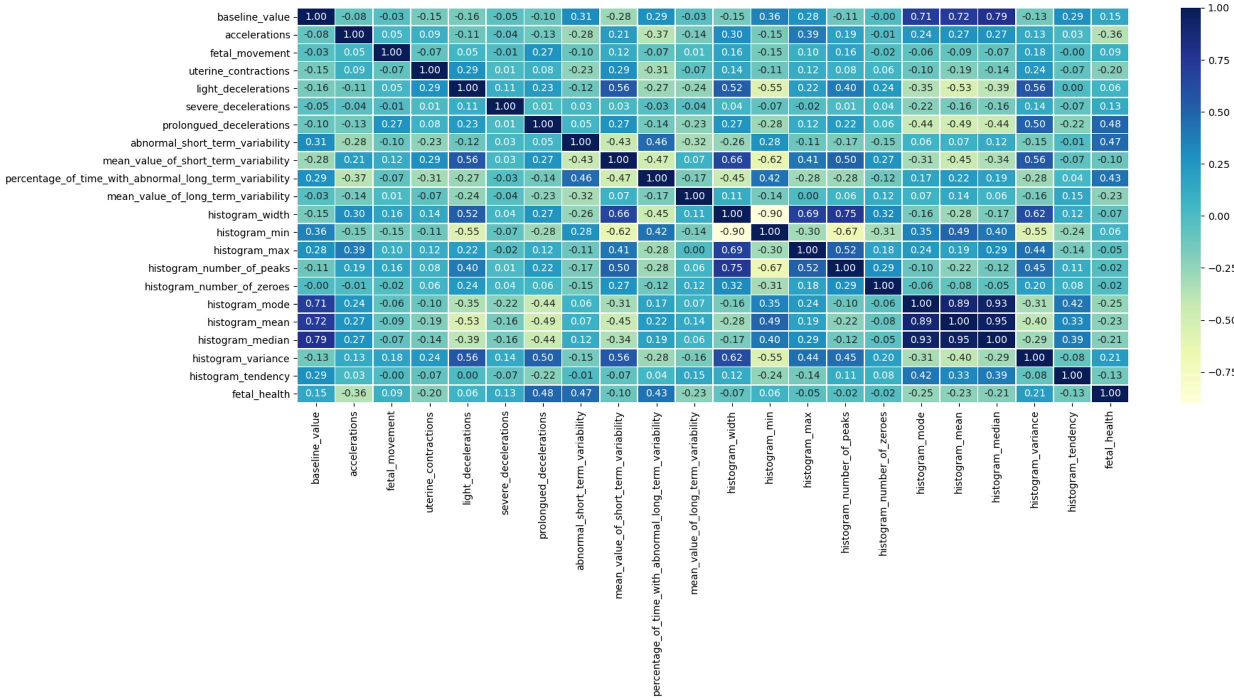

Figure 4: Correlation Matrix of Fetal Health Data

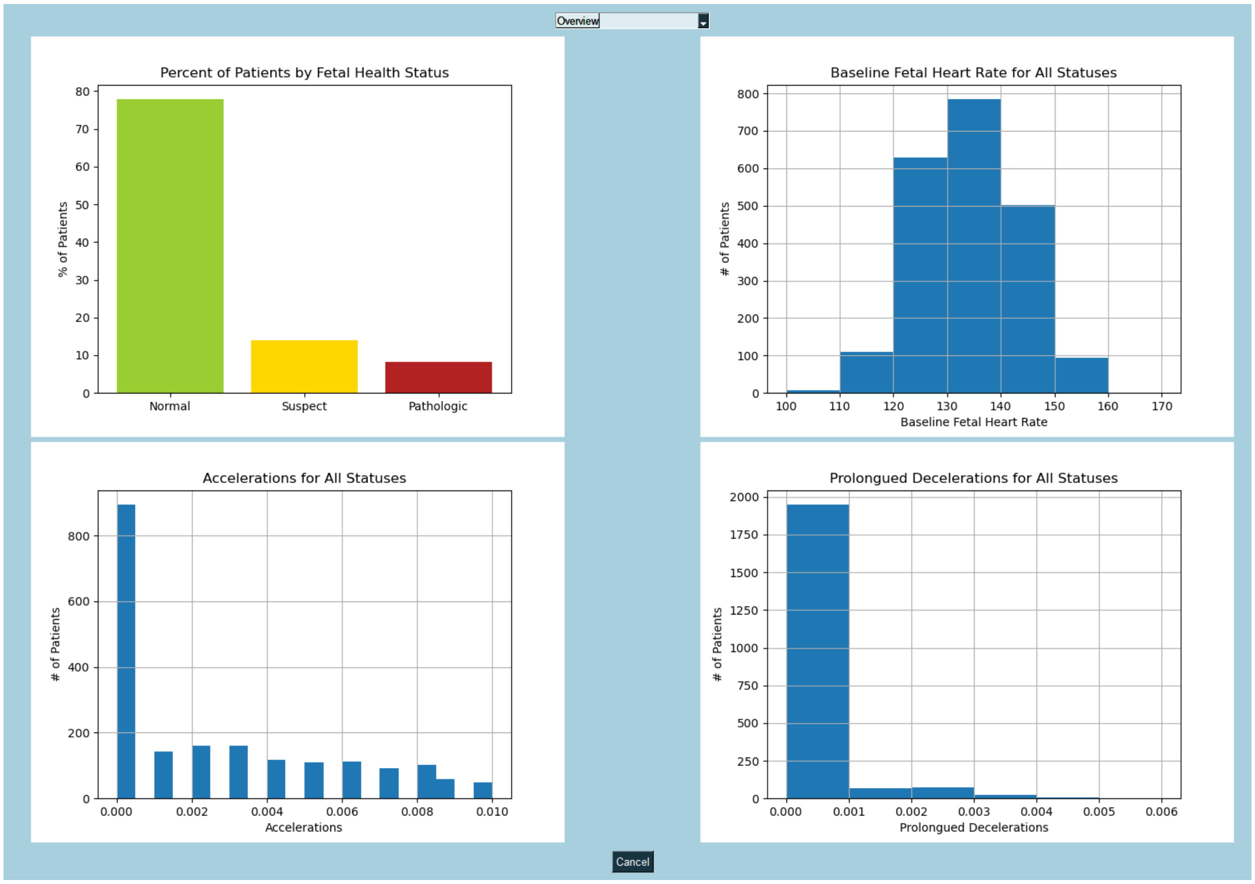

Figure 5: Overall Trends for Features with Strongest Correlations

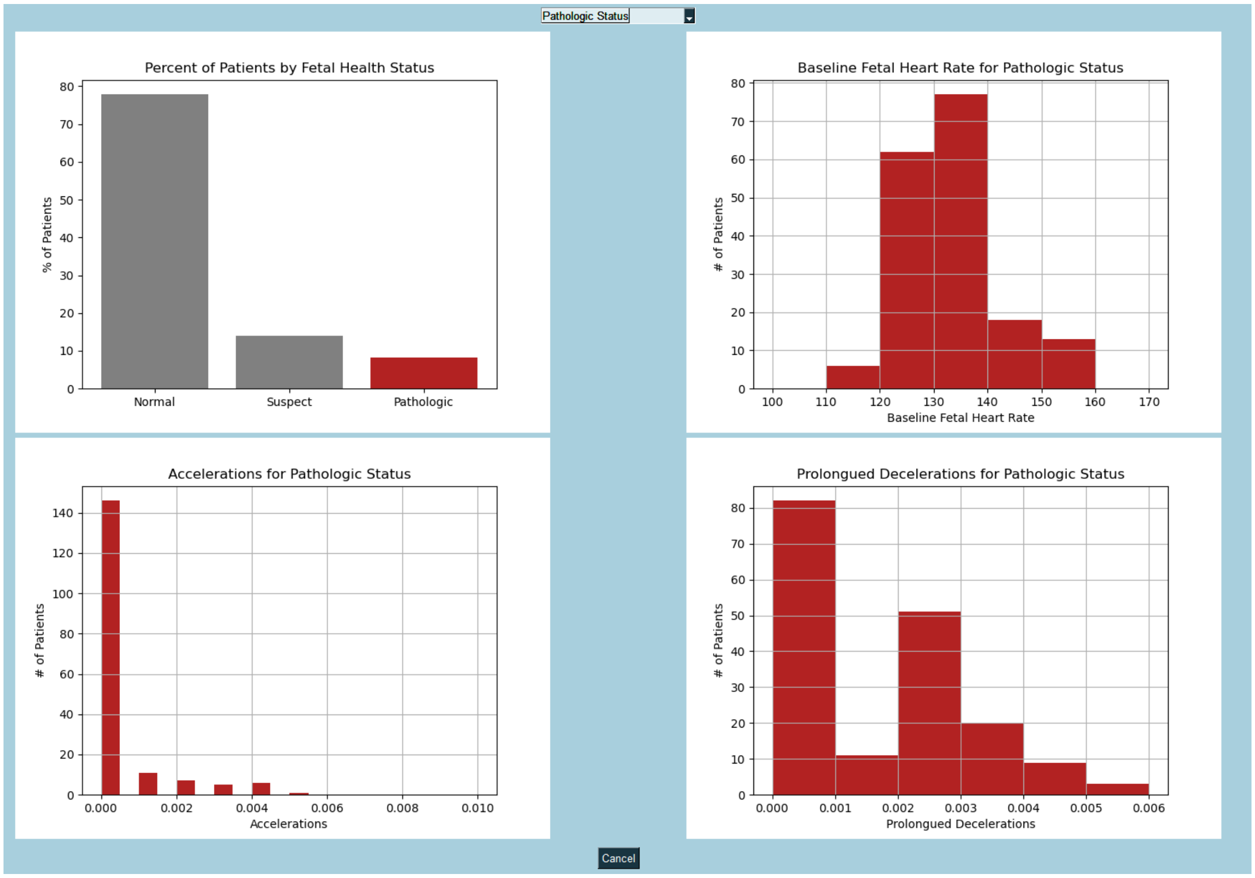

Figure 6: Focusing on Trends in Patients with Pathologic Fetal Health Status

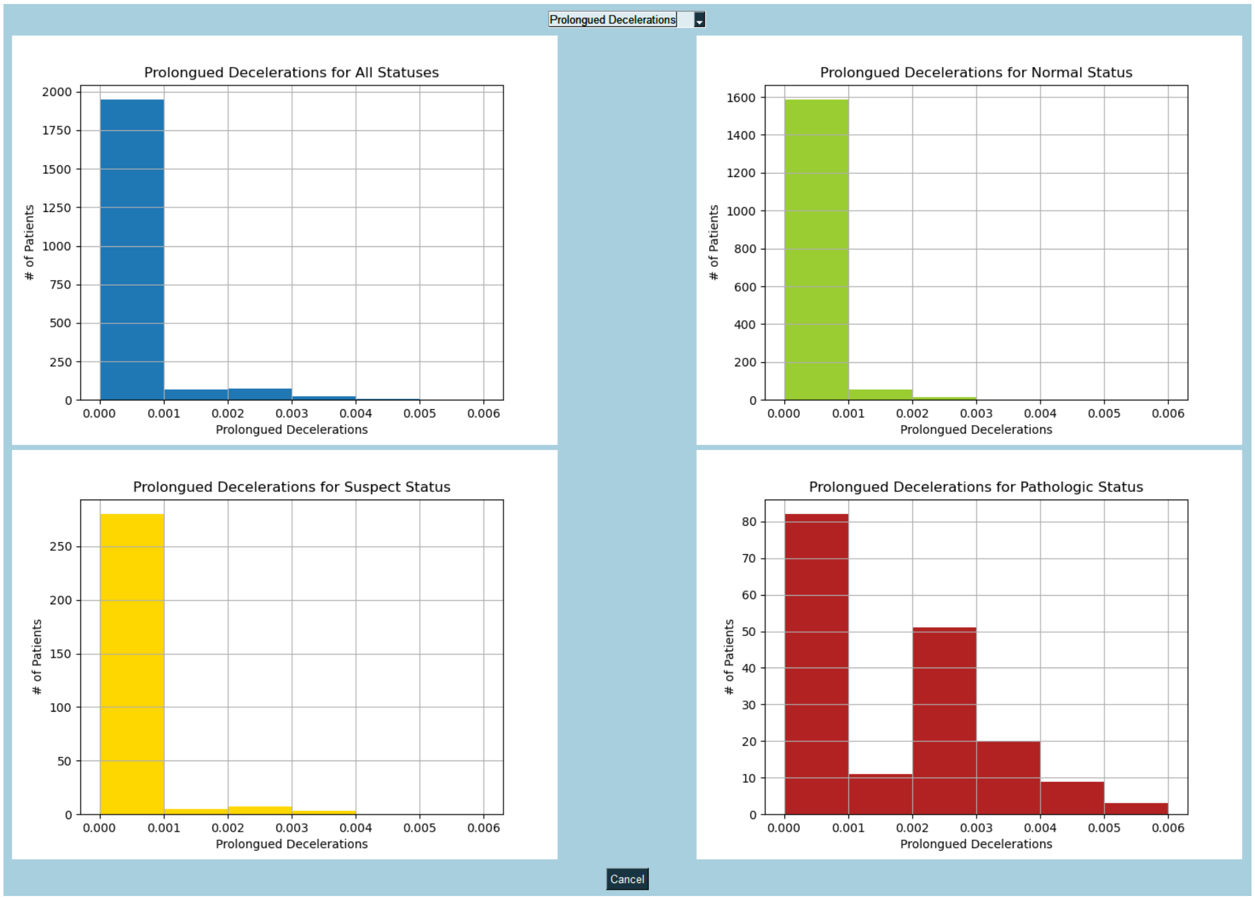

Figure 7: Comparing Prolonged Decelerations Trends Across Fetal Health Status

Accuracy Analysis

Although the model had an accuracy of 95%, the macro F1-score was used to evaluate model performance because both precision and recall for the underrepresented and more critical classifications of suspect and pathologic were deemed more important. The model received a macro F1-score of 92%. It performed best on normal FHS classification and worst on suspect FHS classification. However, it still performed well on pathologic FHS classification, which has the greatest impact on patient wellbeing.

```
Baseline RF

Classifier metrics on the test set
Accuracy: 95.31%
Precision: [0.96735905 0.88333333 0.93103448]
Recall: [0.97897898 0.828125   0.93103448]
F1: [0.97313433 0.85483871 0.93103448]
Classifier metrics on the test set
Macro Precision: 0.9272422888456858
Macro Recall: 0.9127128205791998
Macro F1: 0.919669173598083
```

Figure 8: Model Metrics

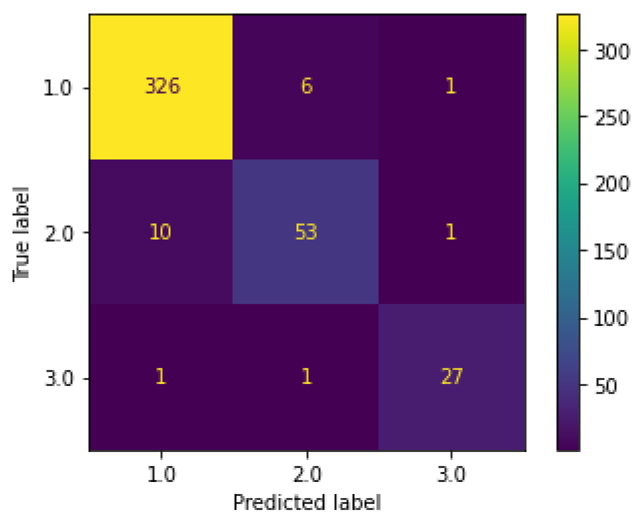

Figure 9: Model Confusion Matrix (Normal = 1.0; Suspect = 2.0; Pathologic = 3.0)

Application Testing

Application testing included unit testing, integration testing, and acceptance testing. During unit testing, each function was tested independently to verify outputs with a variety of inputs that covered both standard and edge cases. During integration testing, each component was tested in conjunction with the other application components. Most of the focus was placed on interactions between the GUI and

SQLite database. Finally, during acceptance testing, end-users from obstetric care teams tested the application to ensure their requirements for functionality, accuracy, and ease of use were met.

Application Files

Table 3: Files Required for Execution

Python Files	Description
main.py	The main entry to the program
dbinter.py	Called by main.py to connect to database
window.py	Called by main.py to control GUI
model.py	Called by window.py to train and update model
JOBLIB Files	
new_model.joblib	The model used by the application
Database Files	
fetal_health_db.db	SQLite database file for fetal health data and user login data

Table 4: Additional Files

File	Description
fetal_health.csv	Dataset from Kaggle
CTG.xls	The original dataset from UCI
fetal-health-data-exploration.ipynb	Jupyter Notebook containing initial data exploration and modeling
fetal-health-data-exploration.html	HTML file containing initial data exploration and modeling (same information as .ipynb file but for quicker/easier viewing)
baseline_rf.joblib	Copy of initial model used by the application (new_model.joblib is overwritten each time a new model is trained)
error_log.txt	Catalogs application errors; created by the application if not found
user_log.txt	Tracks user logins; created by the application if not found

User's Guide

Installation

Prerequisites:

- SQLite 3.34.1
- Python 3.8.5 with the following libraries:
    - Joblib 1.0.0
    - Matplotlib 3.3.2
    - NumPy 1.19.2
    - Pandas 1.2.1
    - PySimpleGUI 4.34.0
    - Python 3.8.5
    - Scikit-learn 0.23.2
    - Seaborn 0.11.1
    - SQLite 3.33.0

Steps:

1. Install the prerequisite applications
2. Extract files from fetal_health.zip
3. Create a new user or skip this step if using the default admin account (username: admin; password: admin)
    a. Open fetal_health_db.db in SQLite
    b. Insert new record in the user table, making sure to memorize username and password as well as set active = 1.
4. Run main.py in Python
5. Example data for input is available in Section E under "Example Fetal Health Data for Testing"

Using the Application

The login screen (Figure 10) will appear upon starting the application. Enter your credentials and click the "Login" button. Once logged in, the menu will appear (Figure 11).

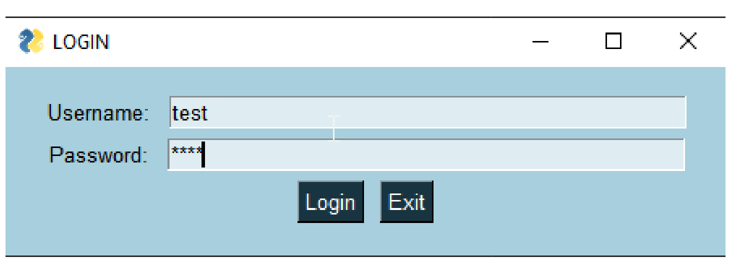

Figure 10: Login Screen

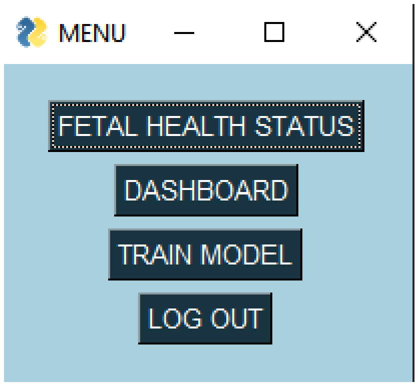

Figure 11: Main Menu

To predict fetal health status or to enter new data into the database, select the “FETAL HEALTH STATUS” button. At this screen (Figure 1), you must enter data into each field before continuing. An error message will appear if the application suspects a typo. After selecting “Calculate”, the predicted fetal health status will appear. To save the data to the database, click “Save to Database”. A new window will appear asking you to confirm the actual health status of the patient. Choose a health status from the dropdown menu and click “Save” to continue. Otherwise, select “Cancel” to return.

To analyze fetal health data, select the “DASHBOARD” button. At this screen (Figure 5), you can select which kind of visualization you would like to access from the dropdown menu at the top of the screen. To return to the main menu, select the “Cancel” button at the bottom of the screen.

To retrain the model on the data, select "TRAIN MODEL" from the menu. On this screen (Figure 12), you can select "TRAIN" to initiate the model training. This may take several minutes to complete. Once complete, a new window with a confusion matrix will appear. This shows the number of predictions for each combination of a predicted label (on the x-axis) and a true label (on the y-axis). Correct predictions will appear in the boxes running diagonally from the upper-left corner to the lower-right corner. Additionally, a report will appear with the model's performance metrics in the previous window. Click the "Save" button to save the model, overwriting the previous model. Alternatively, click the "Cancel" button to keep the original model and return to the main menu.

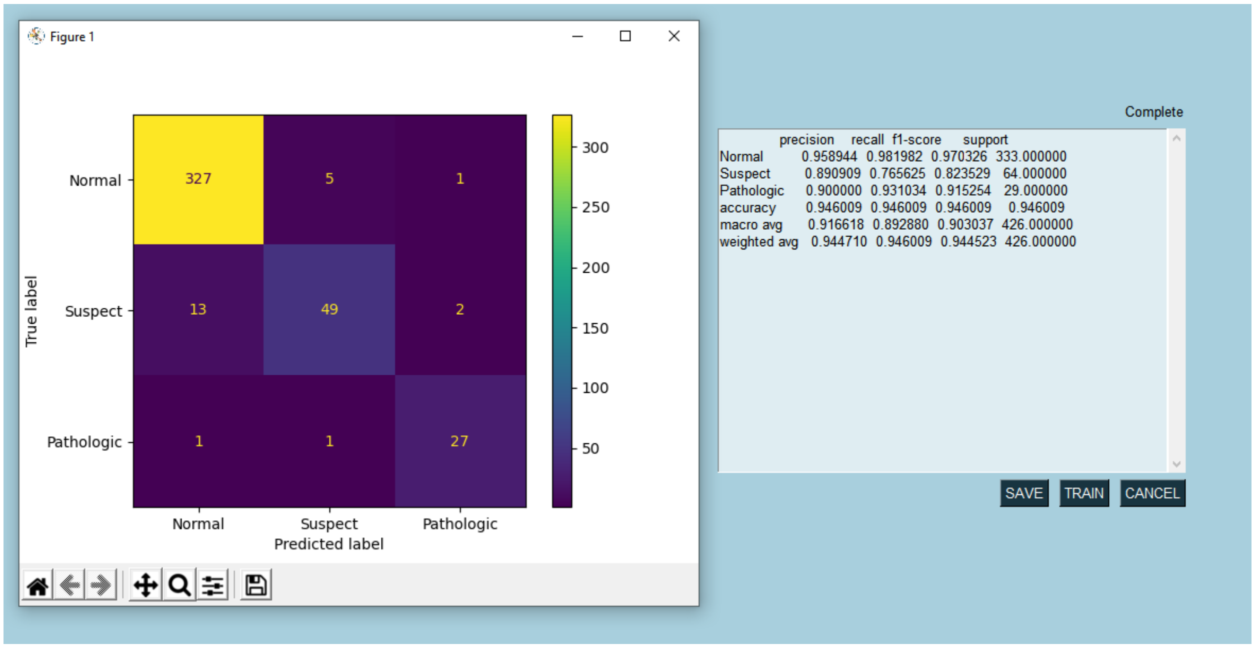

Figure 12: Train Model Screen

When you are finished, you can click the "LOG OUT" button to log out of your account. You can also close out the application by clicking the "X" in the upper right corner of the window.

Summation of Learning Experience

My prior experience with Python was limited, but my initial familiarity made it quicker and easier to start coding and learn new packages. My experience with the PyCharm also made it easier to get started because learning new IDEs can oftentimes slow me down in the beginning. My knowledge of SQL assisted me with setting up the database and its associated tables.

To complete the capstone, I had to learn and practice machine learning techniques, such as data preparation and cleaning, data visualization, data modeling, and hyperparameter tuning. Additionally, I had to learn how to create a GUI in Python.

One of the most challenging components of the capstone was choosing an appropriate dataset and problem. Many datasets and problems were interesting but presented too many challenges for me to complete in time. Some datasets did not have enough data. Others did not have enough features or the right features. And some problems were too complex and would require a longer timeframe to learn the necessary skills.

Overall, this capstone allowed me to develop a product from start to finish. It combined design skills, database management skills, machine learning and algorithm skills, technical writing skills, programming skills, testing and debugging skills, and data analysis and visualization skills all into one project. This experience has encouraged me to step outside of my comfort zone to tackle more difficult problems, explore more interesting datasets, and develop more useful programs.

Section E

Table 5: Example Fetal Health Data for Testing

	Normal	Suspect	Pathologic
Baseline Value	132	120	134
Accelerations	0.006	0	0.001
Fetal Movement	0	0	0
Uterine Contractions	0.006	0	0.01
Light Decelerations	0.003	0	0.009
Severe Decelerations	0	0	0
Prolonged Decelerations	0	0	0.002
Abnormal Short Term Variability	17	73	26
Mean Value of Short Term Variability	2.1	0.5	5.9
Percentage of Time with Abnormal Long Term Variability	0	43	0
Mean Value of Long Term Variability	10.4	2.4	0
Histogram Width	130	64	150
Histogram Min	68	62	50
Histogram Max	198	126	200
Histogram Number of Peaks	6	2	5
Histogram Number of Zeroes	1	0	3
Histogram Mode	141	120	76
Histogram Mean	136	137	107
Histogram Median	140	121	107
Histogram Variance	12	73	170
Histogram Tendency	0	1	0

References

Ayres-de-Campos, D., Bernardes, J., Garrido, A., Marques-de-Sá, J., & Pereira-Leite, L. (2000). SisPorto 2.0: A program for Automated Analysis of Cardiotocograms. *Journal of Maternal-Fetal and Neonatal Medicine*, 9(5), 311-318. doi:10.3109/14767050009053454

Chandraharan, E., & Arulkumaran, S. (2007). Prevention of birth asphyxia: Responding appropriately to cardiotocograph (ctg) traces. *Best Practice & Research Clinical Obstetrics & Gynaecology*, 21(4), 609-624. doi:10.1016/j.bpobgyn.2007.02.008

Dua, D., & Graff, C. (2019). UCI Machine Learning Repository: Cardiotocography Data Set. Retrieved February 23, 2021, from https://archive.ics.uci.edu/ml/datasets/cardiotocography

Maranhão, A. (2020, October 12). Fetal Health Classification. Retrieved February 23, 2021, from https://www.kaggle.com/andrewmvd/fetal-health-classification

Murphy, K. W., Johnson, P., Moorcraft, J., Pattinson, R., Russell, V., & Turnbull, A. (1990). Birth asphyxia and the intrapartum cardiotocograph. *BJOG: An International Journal of Obstetrics and Gynaecology*, 97(6), 470-479. doi:10.1111/j.1471-0528.1990.tb02515.x