Princess Sumaya University for Technology

King Hussein School for Computing Sciences

Fetal Health Detection System

Prepared By:

Leen Abu Shaar
20170072

Osama Kuri
20160078

Software Engineering Course Project

Course Number: 11425

Second Semester - 2020/2021

The Github repository

https://github.com/The-Osk/Fetal_Health_classification

The Web application

https://share.streamlit.io/the-osk/fetal_health_classification/main/App.py

List of Abbreviations

WHO: World Health Organization

ML: Machine Learning

CTGs: Cardiotocograms

Table of Contents

Table of Figures

Table of Tables

Chapter 1: Introduction

1.1 Introduction

Fetal health and well-being is a major concern that needs constant and accurate examination throughout pregnancies. The science behind fetal health is an ever-lasting topic of research where research centers, researchers and medics have dedicated time to because of its medical importance and humanitarian value. Since there are vast causes of complications, identification becomes a challenge that has to be taken seriously in order to maintain healthy pregnancies.

Fetal and maternal mortality rate trends have been observed over the years, and the World Health Organization (WHO) has concluded a decrease in these rates (Woods). Advancements in obstetrics including medical assessments and equipment have surely had an impact in decreasing mortality rates. In addition, technological solutions like the development of point-of-care testing through wearable and mobile devices have been introduced (Shrivastava et al.). With the advances carried out by the field of machine learning (ML) and the impact of its applications on the healthcare industry, it is only relevant to look at ML solutions that aim to drop these rates even further.

ML applications in healthcare provide insightful guidance to professionals in the field, and act as a coexisting and powerful tool to diagnose and assess. The system in which this project sheds light on is to be used by obstetricians in clinics and hospitals. The information obtained by fetal monitoring devices will act as the input data for the system, including fetal heart rate and fetal movements per second. The features are selected based on collected medical data of the correlations between these measurements and fetal health, in hope that it will provide an accurate insight on the output of the diagnosis.

1.2 System Description

The system will be an application that the user can enter data collected from Cardiotocograms (CTGs) for a patient and a trained ML model will predict if the patient is healthy, suspected to have health problems, or has a fatal one.

1.3 System Purpose

This system aims to make the process of checking on the wellbeing of fetuses and mothers easier for hospitals and clinics, while maintaining good accuracy.

1.4 Problem Statement

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.

To achieve such goals we need better systems to predict the fatal health risks and using the power of machine learning can help achieve this.

1.5 The System Context View

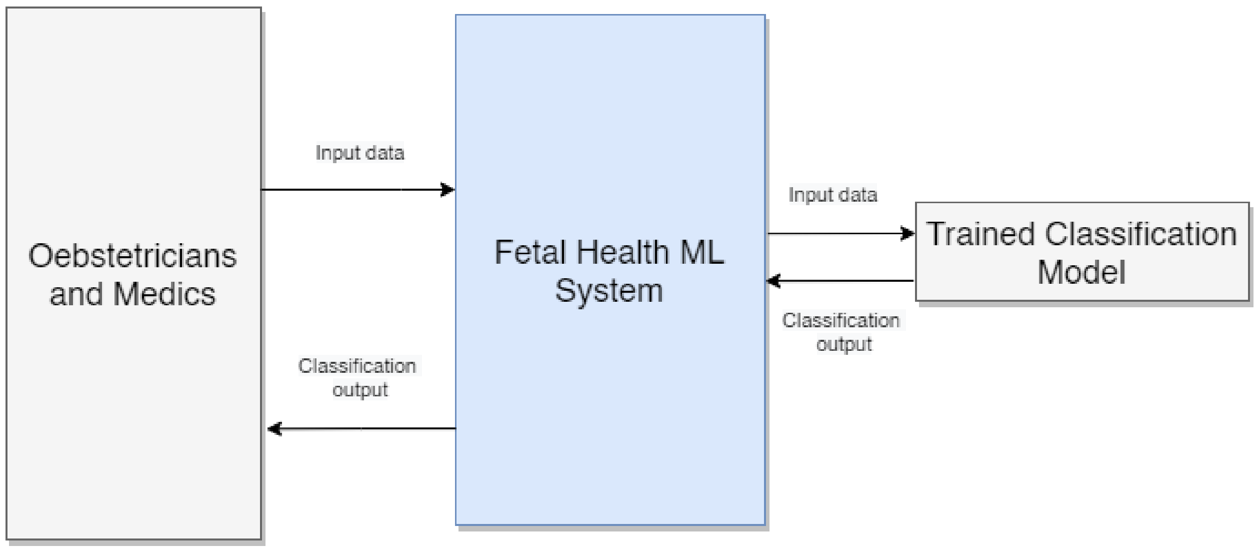


Figure 1: The System Context View

Shows the interaction between health professionals, the system, and the model it uses for prediction.

1.6 Literature Review

1.6.1 "Classification Fetal Health" Work [1]

In this work, the same dataset of this paper was used. Only two machine learning models were implemented. These machine learning models are: Logistic Regression and Support Vector Machine. The accuracy of the LR was 84% while in SVM it was 91% with a Mean Square Error (MSE) of 0.1.

1.6.2 "4 Fetal Health Classifiers using GridSearchCV" Work [2]

For the same dataset as well, the work in paper [7] used four machine learning models: K Nearest Neighbors, Logistic Regression, Support Vector Machine and Random Forest. The highest accuracy was achieved by the Random Forest model by a rate of 93.4%.

1.6.3 “Comparison of Machine Learning Techniques for Fetal Heart Rate Classification” Paper [3]

This IEEE report discusses which machine learning model is the most efficient to classify a fetal heart rate signal into 1) normal, 2) hypoxic. The machine learning techniques used were: artificial neural network, support vector machine, extreme learning machine, radial basis function network, and random forest. All techniques showed acceptable results but the best outcome of accuracy 97.94% was achieved by using the artificial neural network.

1.6.4 “Use of Machine Learning Algorithms for Prediction of Fetal Risk using Cardiotocographic Data” Paper [4]

The work presented in paper [4] used a different dataset. However, their dataset was obtained from the machine learning datasets repository of the University of California. similarly to the machine learning system implemented in this paper, the work in [4] discusses 10 different classification models to predict normal, suspect, and pathological fetal states. The model with the best overall accuracy rate was the XGBoost with a rate of 96%.

1.7 Challenges

- High Accuracy:

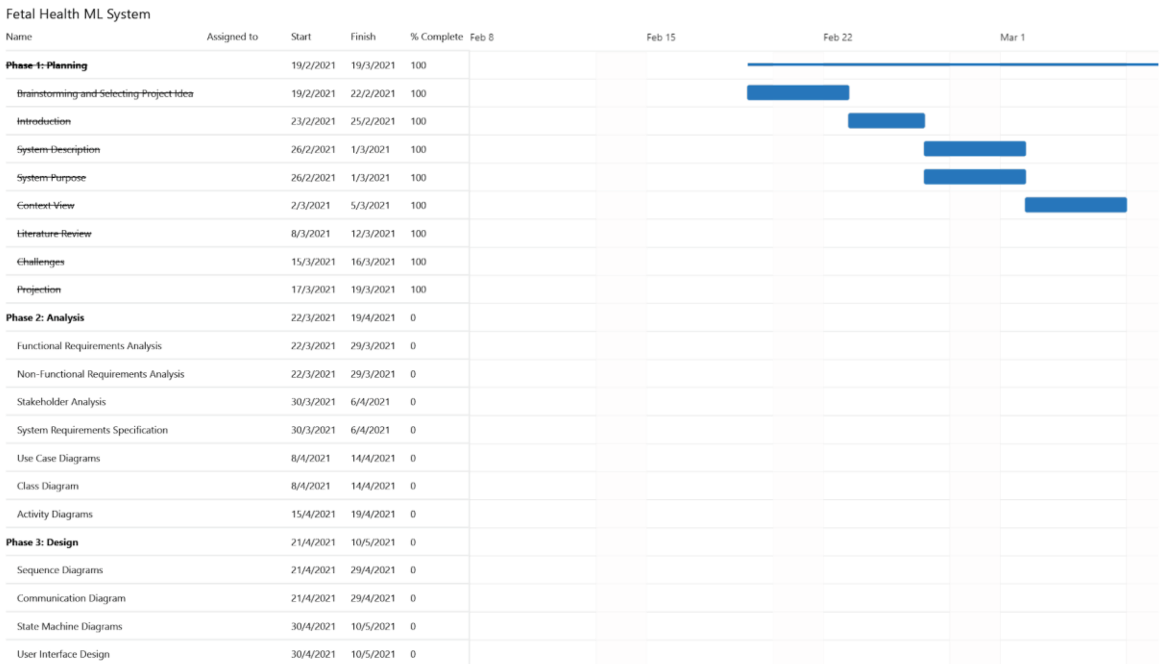
  High accuracy is important for such a system, but a high accuracy cannot always be granted.

- Limitation of Data:

  Machine learning requires a lot of data which can be hard to find, we are using data found in Kaggle [5] as a baseline with the hope that more data is gathered from users to improve the model.

1.8 Projection

1.8.1 Gantt Chart and Tasks



Fetal Health ML System

Name	Assigned to	Start	Finish	% Complete
~~**Phase 1: Planning**~~		19/2/2021	19/3/2021	100
~~Brainstorming and Selecting Project Idea~~		19/2/2021	22/2/2021	100
~~Introduction~~		23/2/2021	25/2/2021	100
~~System Description~~		26/2/2021	1/3/2021	100
~~System Purpose~~		26/2/2021	1/3/2021	100
~~Context View~~		2/3/2021	5/3/2021	100
~~Literature Review~~		8/3/2021	12/3/2021	100
~~Challenges~~		15/3/2021	16/3/2021	100
~~Projection~~		17/3/2021	19/3/2021	100
Phase 2: Analysis		22/3/2021	19/4/2021	0
Functional Requirements Analysis		22/3/2021	29/3/2021	0
Non-Functional Requirements Analysis		22/3/2021	29/3/2021	0
Stakeholder Analysis		30/3/2021	6/4/2021	0
System Requirements Specification		30/3/2021	6/4/2021	0
Use Case Diagrams		8/4/2021	14/4/2021	0
Class Diagram		8/4/2021	14/4/2021	0
Activity Diagrams		15/4/2021	19/4/2021	0
Phase 3: Design		21/4/2021	10/5/2021	0
Sequence Diagrams		21/4/2021	29/4/2021	0
Communication Diagram		21/4/2021	29/4/2021	0
State Machine Diagrams		30/4/2021	10/5/2021	0
User Interface Design		30/4/2021	10/5/2021	0

Figure 2: Project's Gantt Chart for the Months of February and March

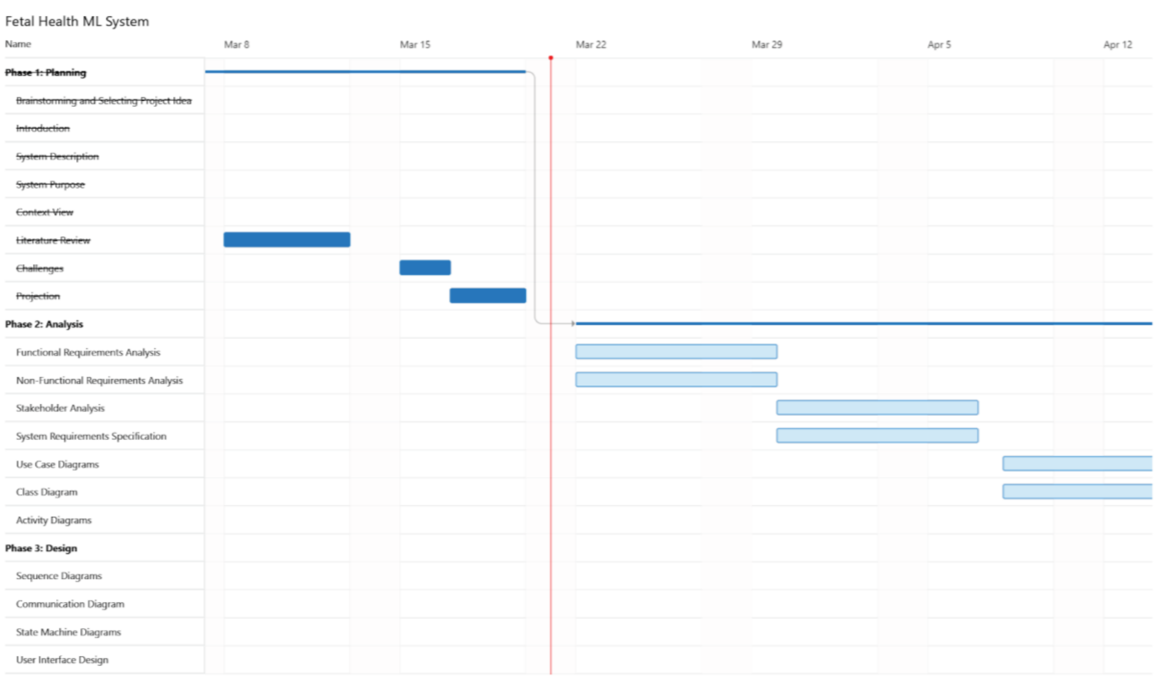


Figure 3: Project's Gantt Chart for the Months of March and April

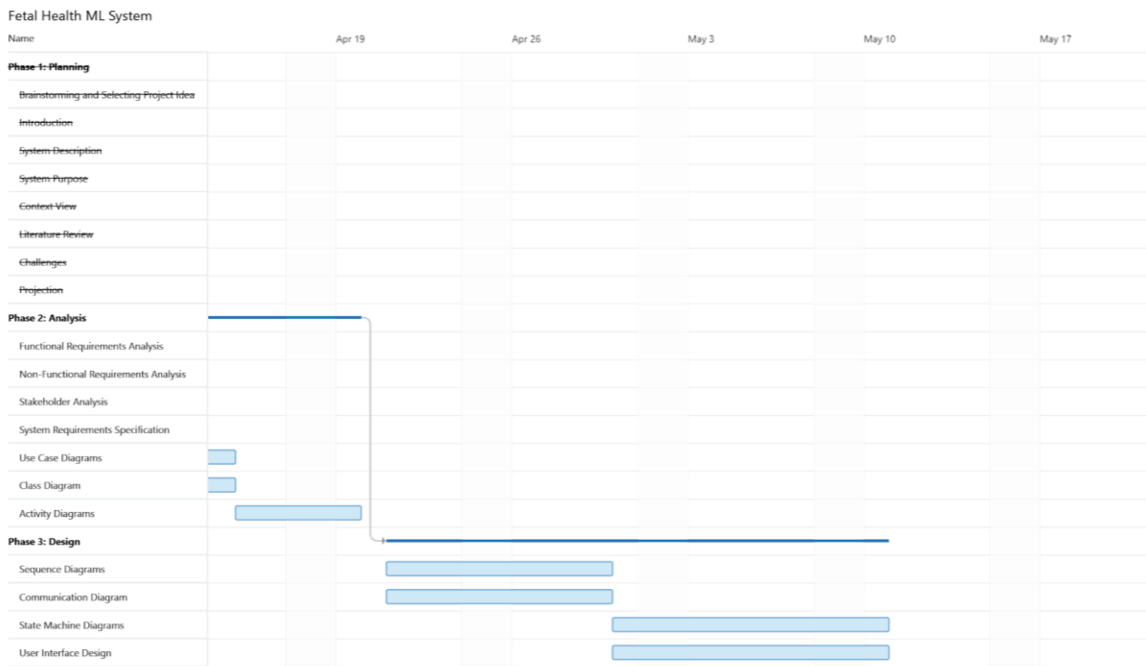


Figure 4: Project's Gantt Chart for the Months of April and May

Activity	**Predecessors**	**Duration (days)**
Brainstorming and Selecting Project Idea	-	2
Introduction	-	3
System Description	Introduction	2
System Purpose	Introduction	2
Context View	System Description, System Purpose	4
Literature Review (Researching Similar Fetal Health Machine Learning Systems)	Context View	5
Challenges	Context View	2
Projection	Context View	3

Table 1: Phase 1 Tasks

Activity	Predecessors	Duration (days)
Functional Requirements	-	4
Non-Functional Requirements	Functional Requirements	5
Class Diagram	Non-Functional Requirements	10
Use Case	Non-Functional Requirements	8
Activity Diagram	Non-Functional Requirements	5

Table 2: Phase 2 Tasks

Activity	Predecessors	Duration (days)
Sequence Diagram	-	2
Communication Diagram	-	5
State Machine Diagram	-	5
User Interfaces	-	10
Conclusion and Future work	User Interfaces	4

Table 3: Phase 3 Tasks

1.8.2 Network Diagram

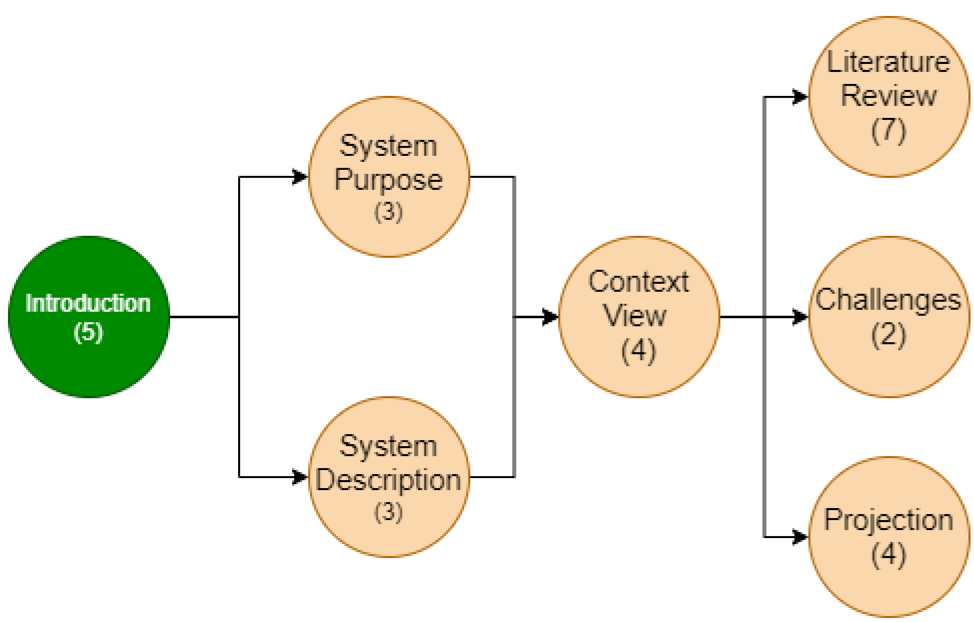


Figure 5: Phase 1 Network Diagram

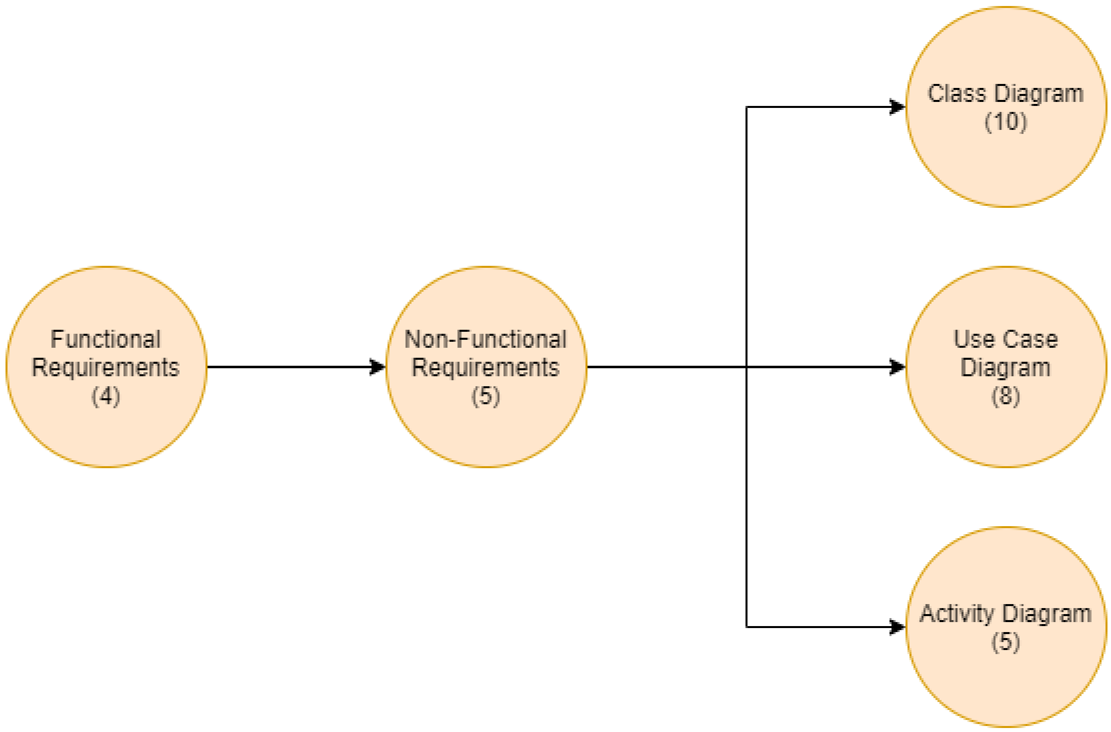


Figure 6: Phase 2 Network Diagram

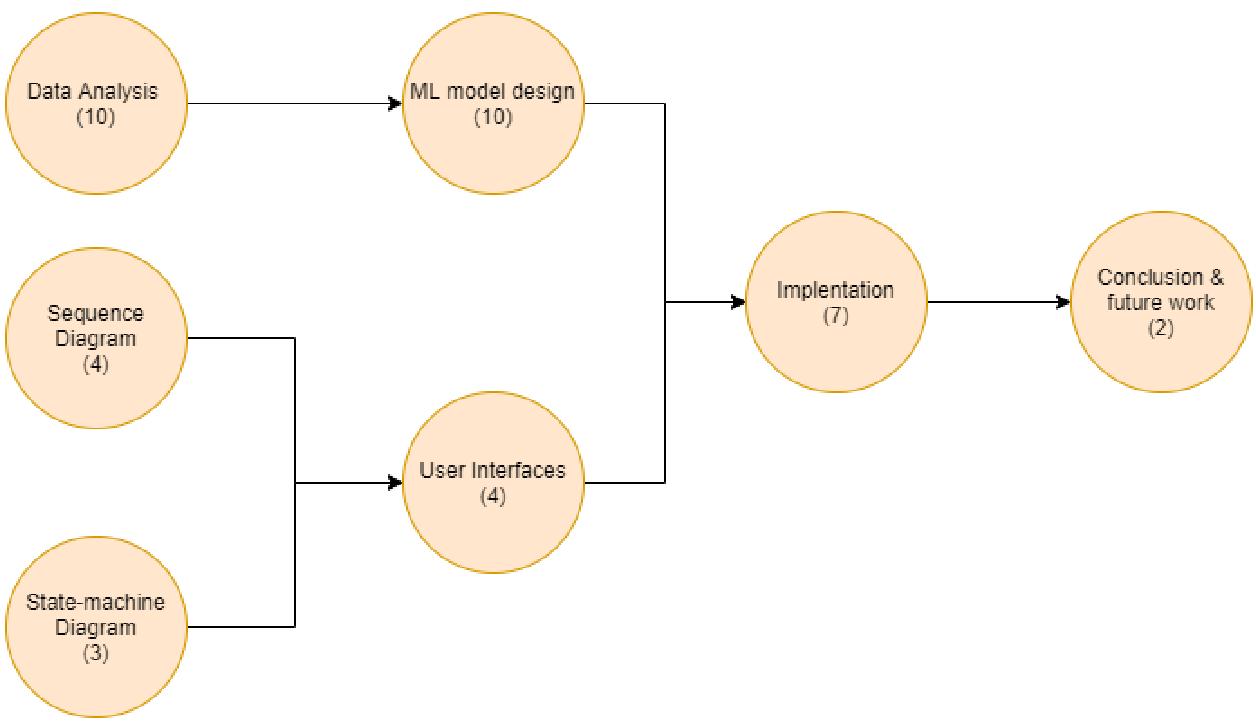


Figure 7: Phase 3 Network Diagram

Chapter 2: Analysis and Design

2.1 Functional Requirements of the System

Requirement ID	Requirement Name	Requirement Detail
FR1	Predict Fetal Health	User inputs the patient's medical data to the system and the system predicts fetal health and displays output

Table 4: Functional Requirements

2.2 Non-Functional Properties of the System

Requirement ID	Requirement Name	Requirement Detail
NFR1	Reliability	The model should give accurate results.
NFR2	Usability	The system should be easy to use by anyone.
NFR3	Performance	The system should give on-the-spot results.

Table 5: Non-Functional Requirements

2.3 Class Diagram

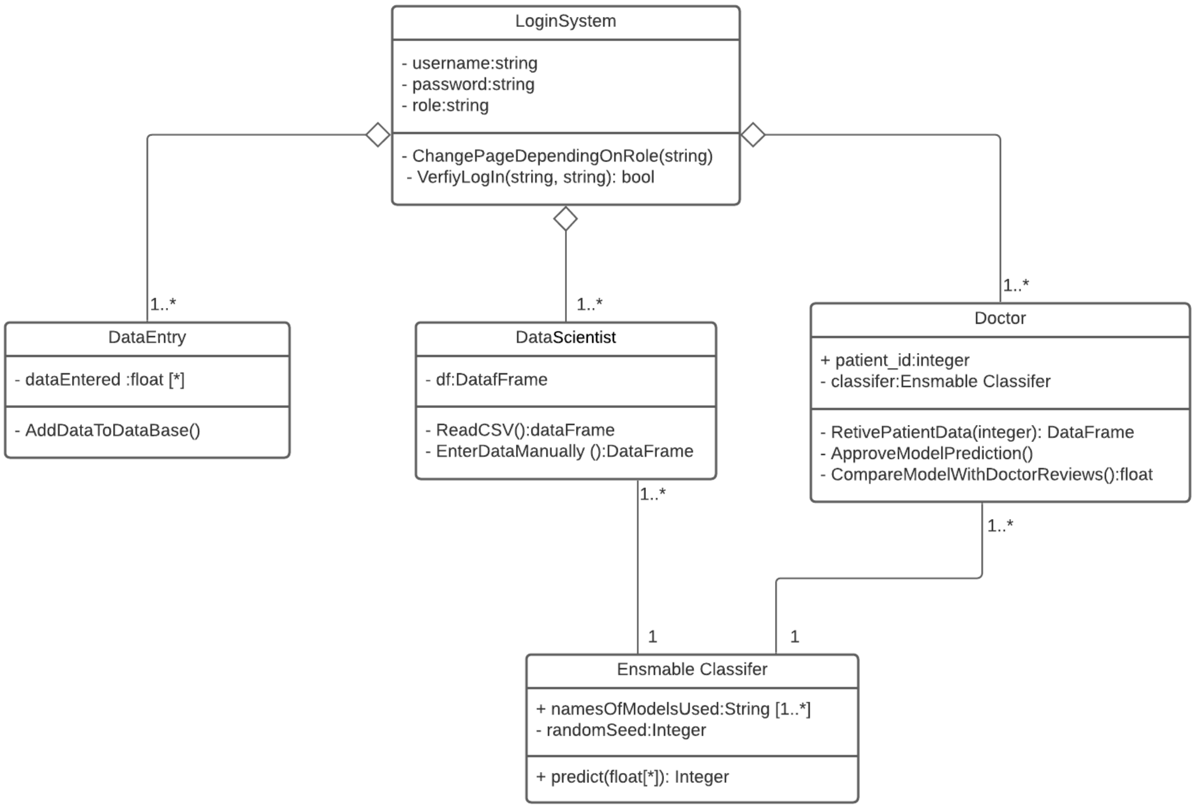


Figure 8: Class Diagram

Showing the relations between the classes used.

2.4 Use Cases

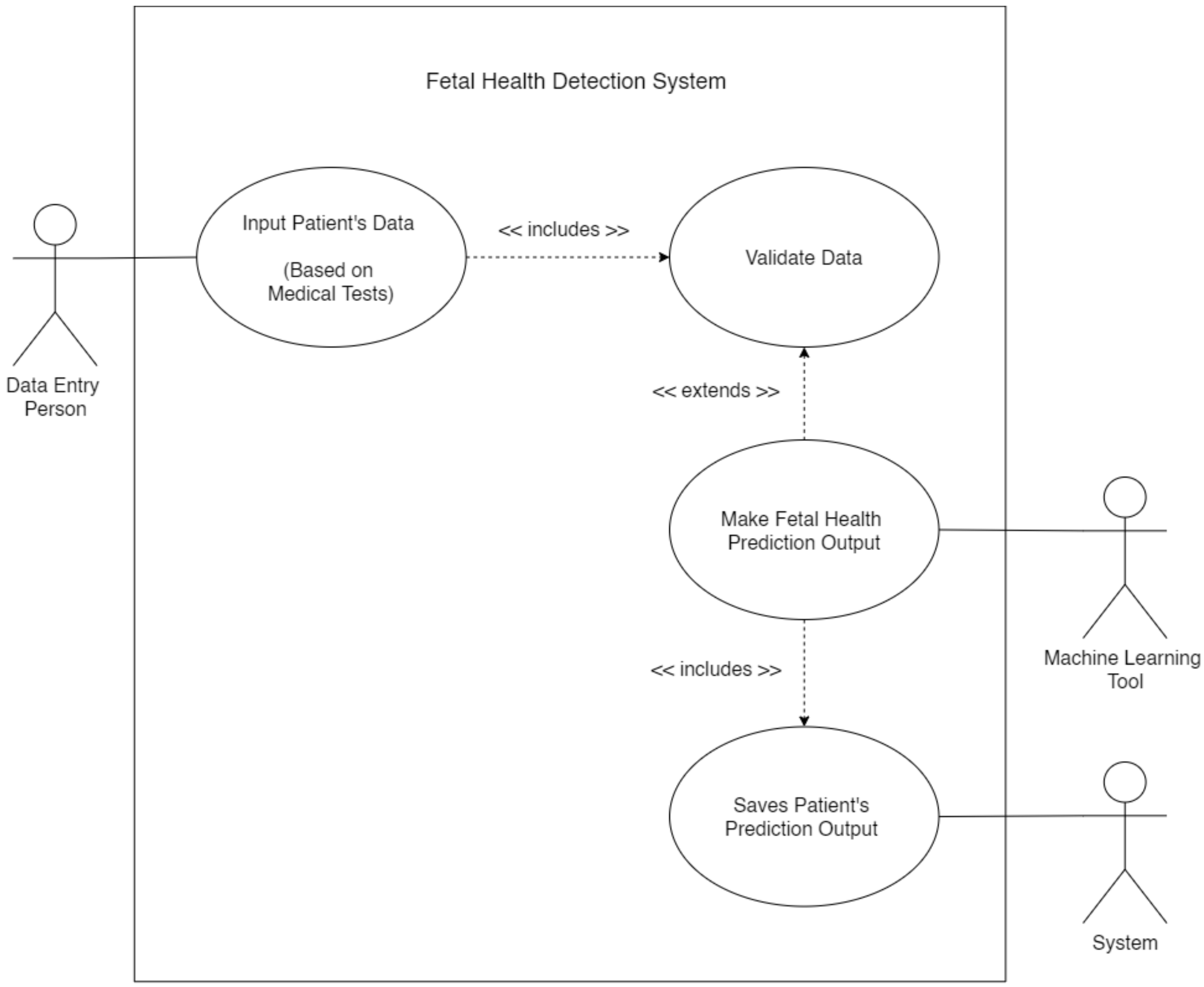


Figure 9: Use Case Diagram 1

Showing the use cases related to the person entering the patient’s data, and these use cases’ interaction with the machine learning tool and the system.

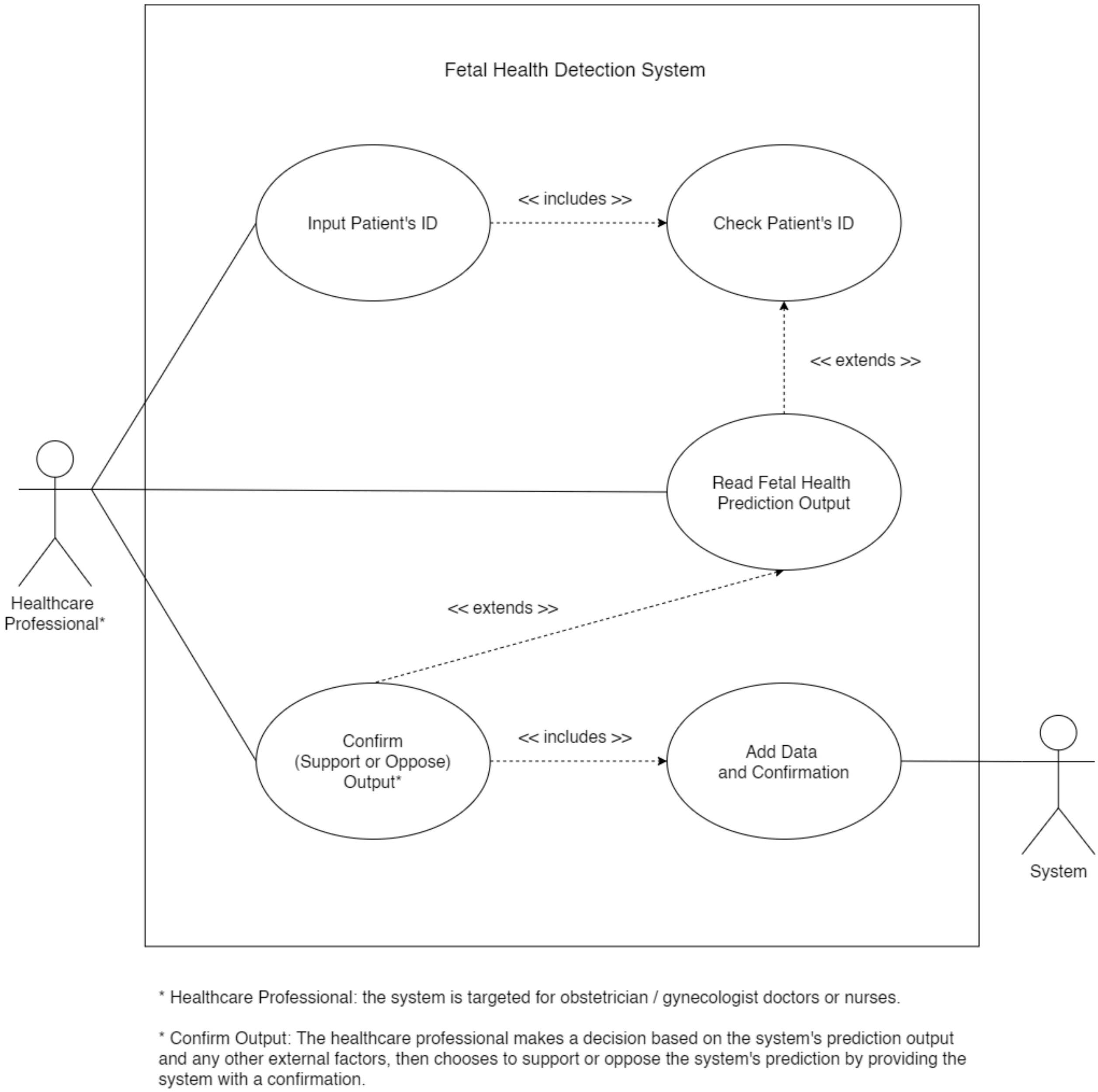


Figure 10: Use Case Diagram 2

Showing the use cases related to the healthcare professional entering the patient's ID, and these use cases' interaction with the system.

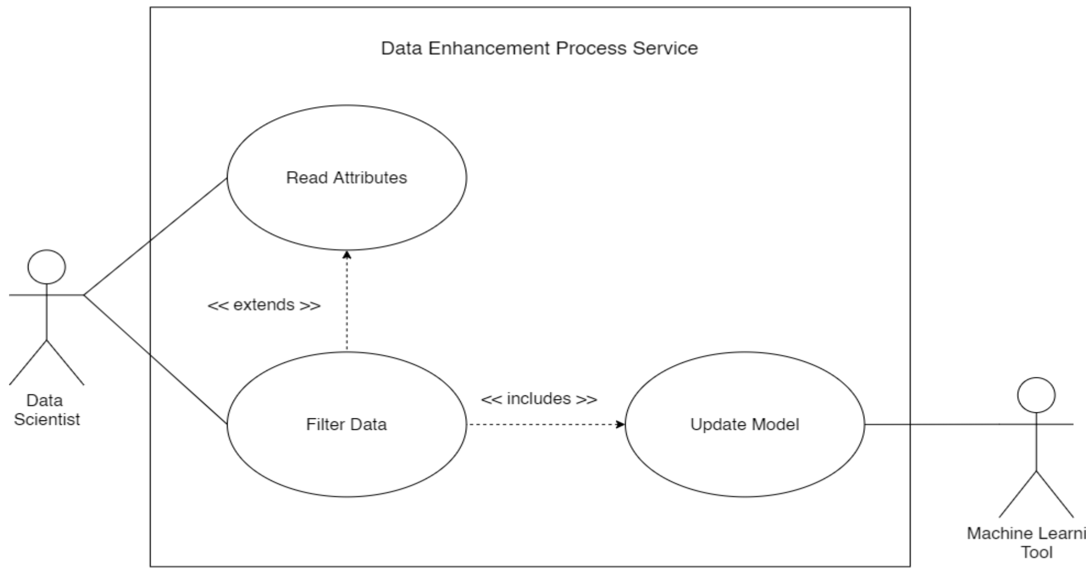


Figure 11: Use Case Diagram 3

Showing the use cases related to the data scientist of the system, and these use cases' interaction with the machine learning tool used by the system.

2.5 Activity Diagram

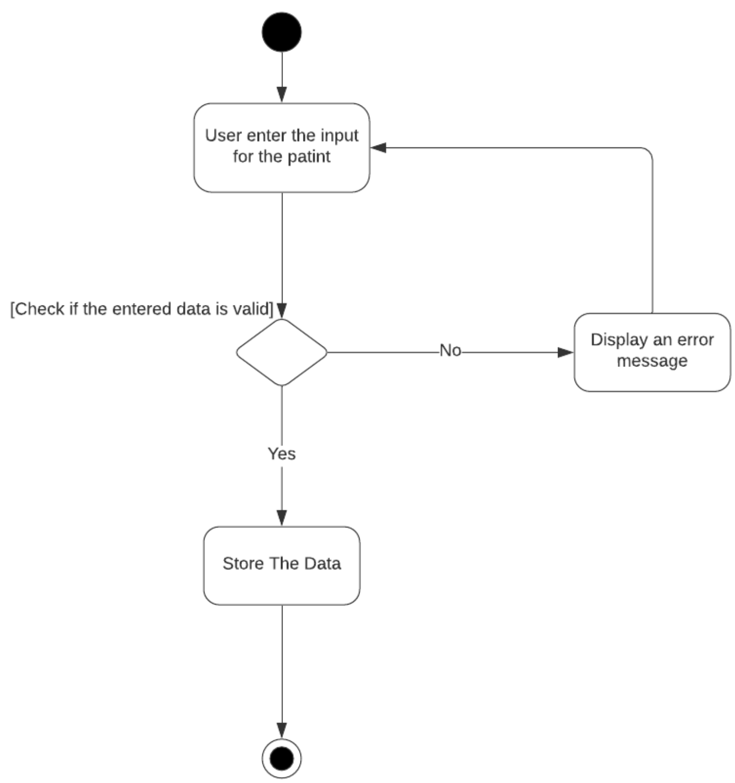


Figure 12: Activity Diagram 1

Describing the activity of entering the patient's data to the system.

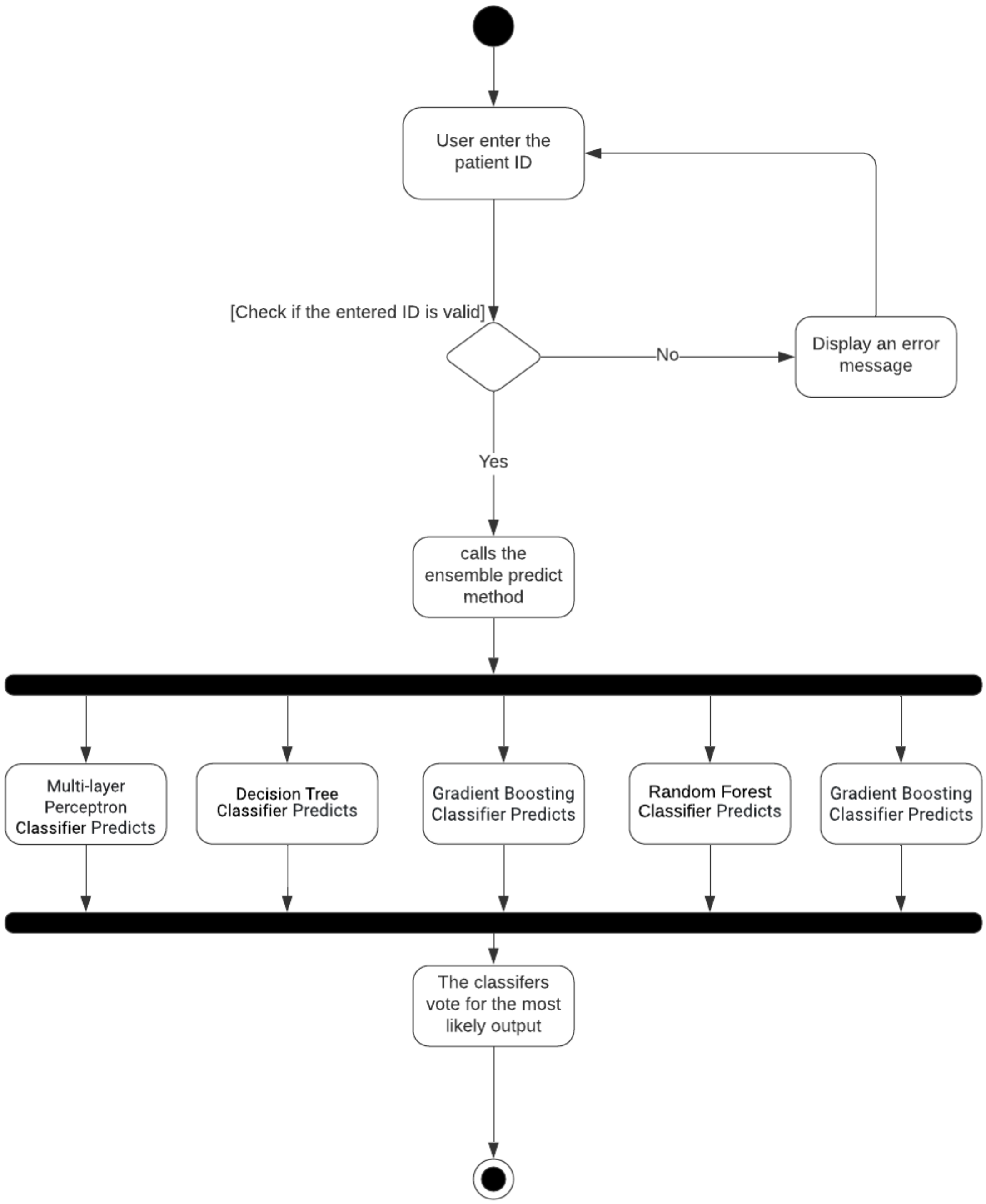


Figure 13: Activity Diagram 2

Describing the activity of entering the patient's ID to the system in order to input their data to the machine learning tool and retrieve an accurate prediction output to the user.

Chapter 3: Analysis and Design

3.1 Sequence Diagram

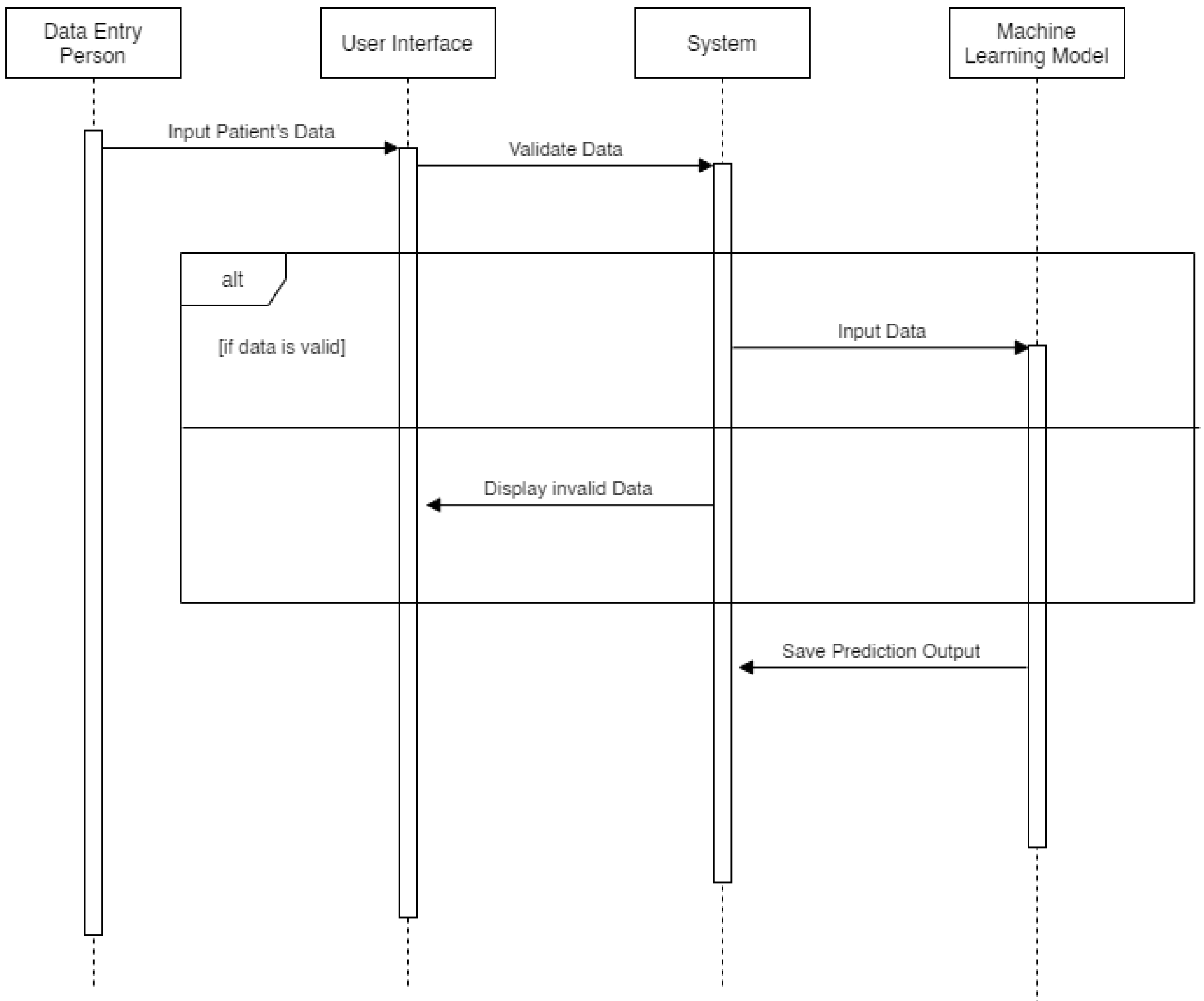


Figure 14: Sequence Diagram 1

Sequence diagram for the data entry.

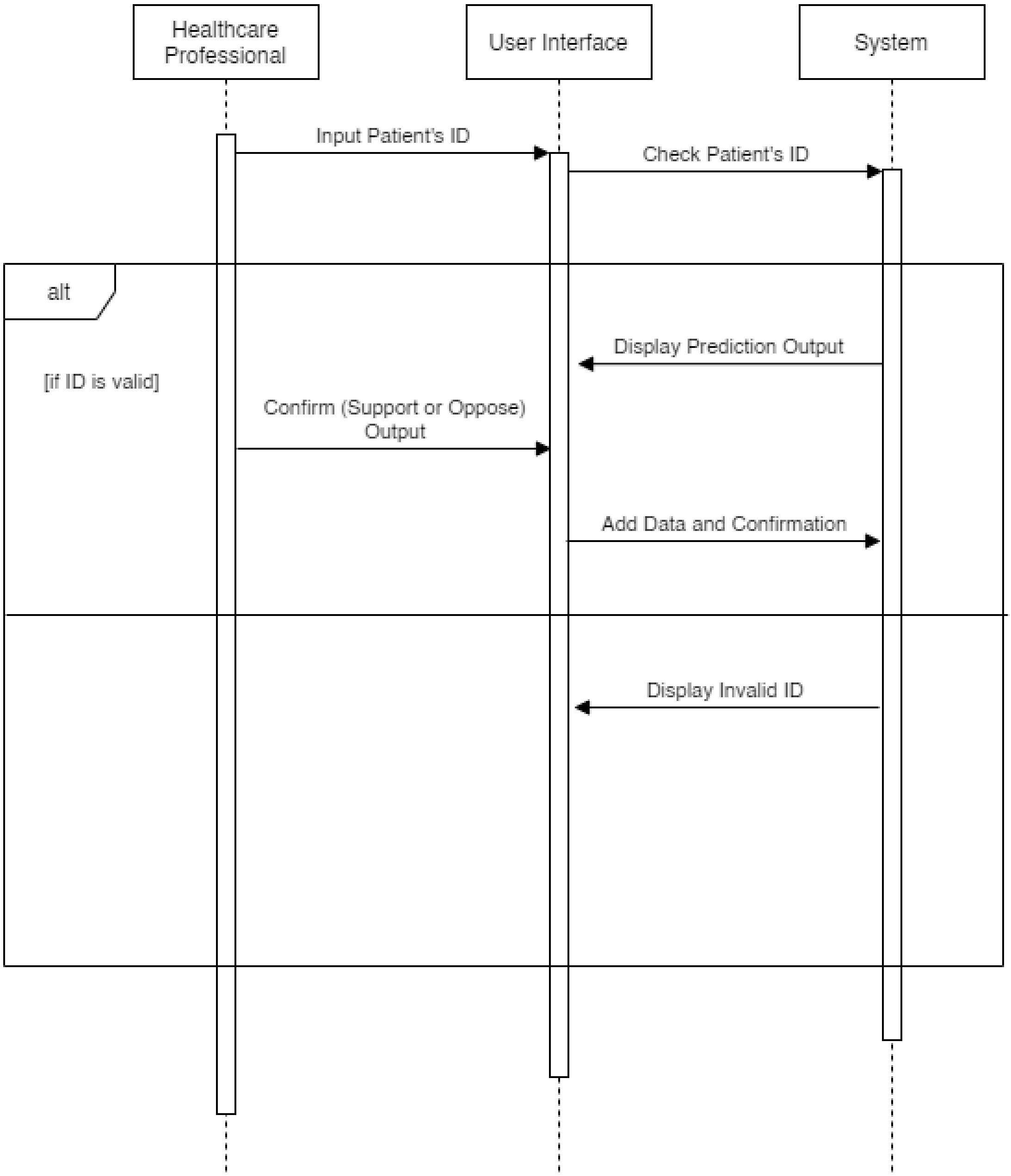


Figure 15: Sequence Diagram 2

Sequence diagram for showing prediction output.

3.2 State Machine Diagrams

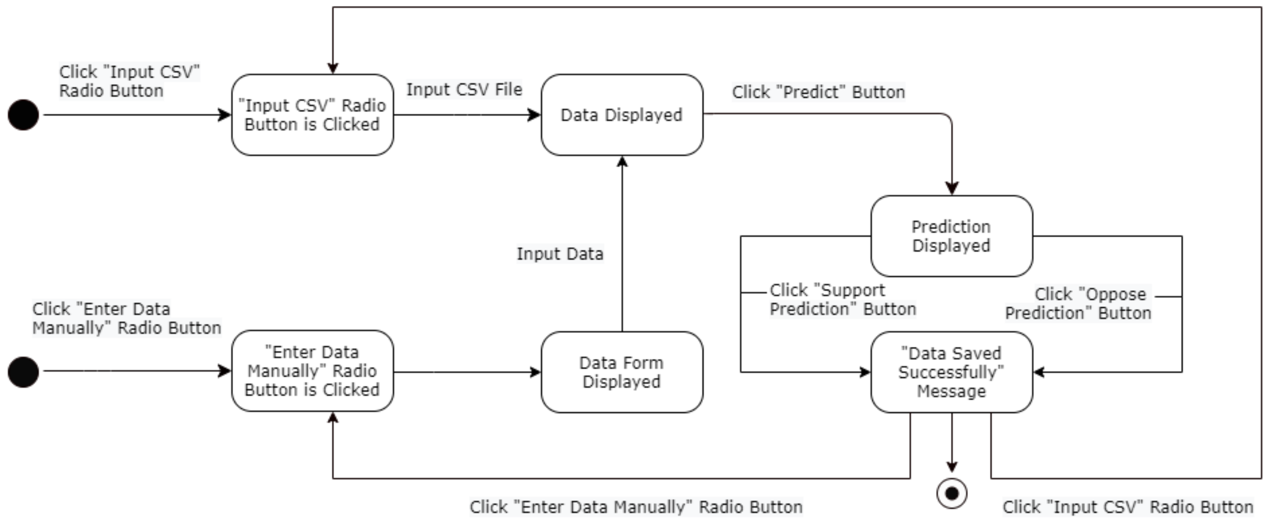


Figure 16: State Machine Diagram 1

Showing state machine diagram for user interface states.

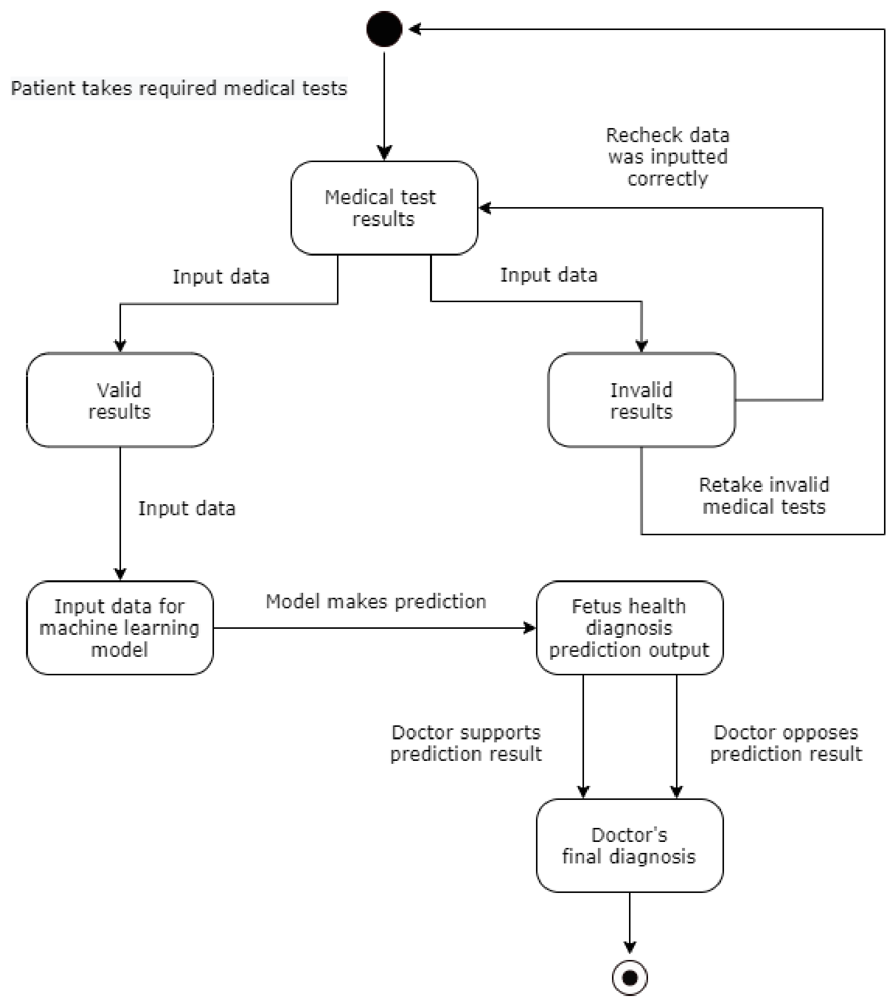


Figure 17: State Machine Diagram 2

Showing state machine diagram for patient's diagnosis.

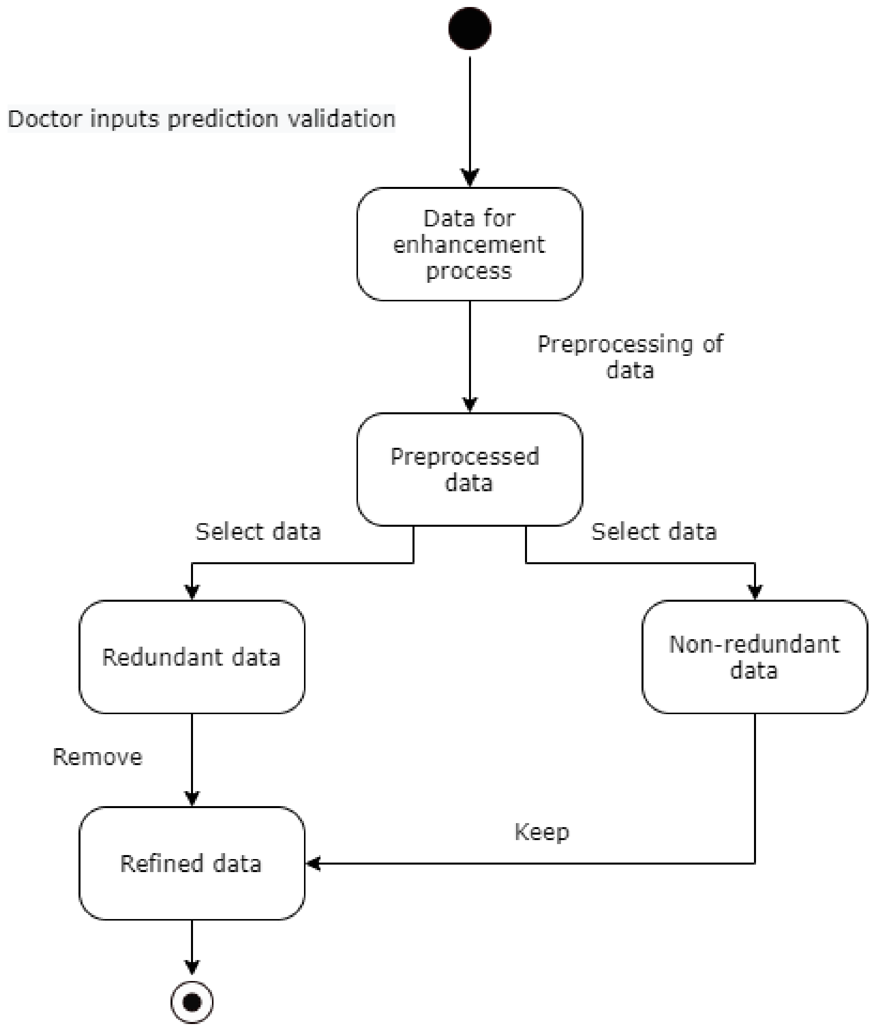


Figure 18: State Machine Diagram 3

Showing state machine diagram for the data enhancement process.

3.3 User Interfaces

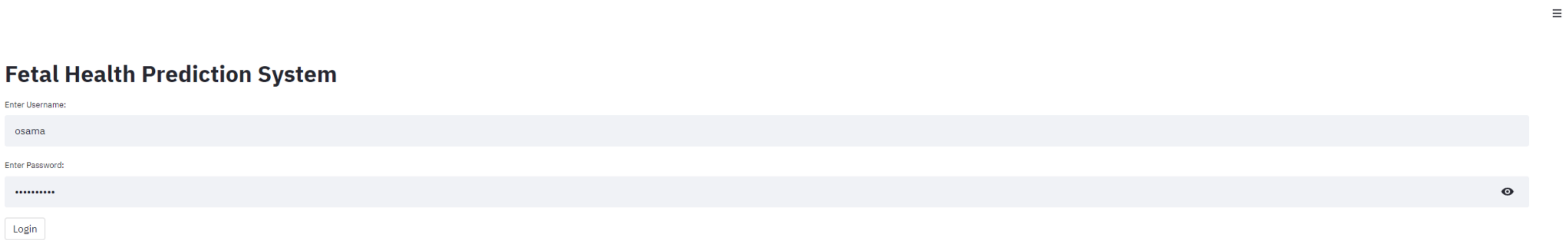


Figure 19: The Login Page User Interface

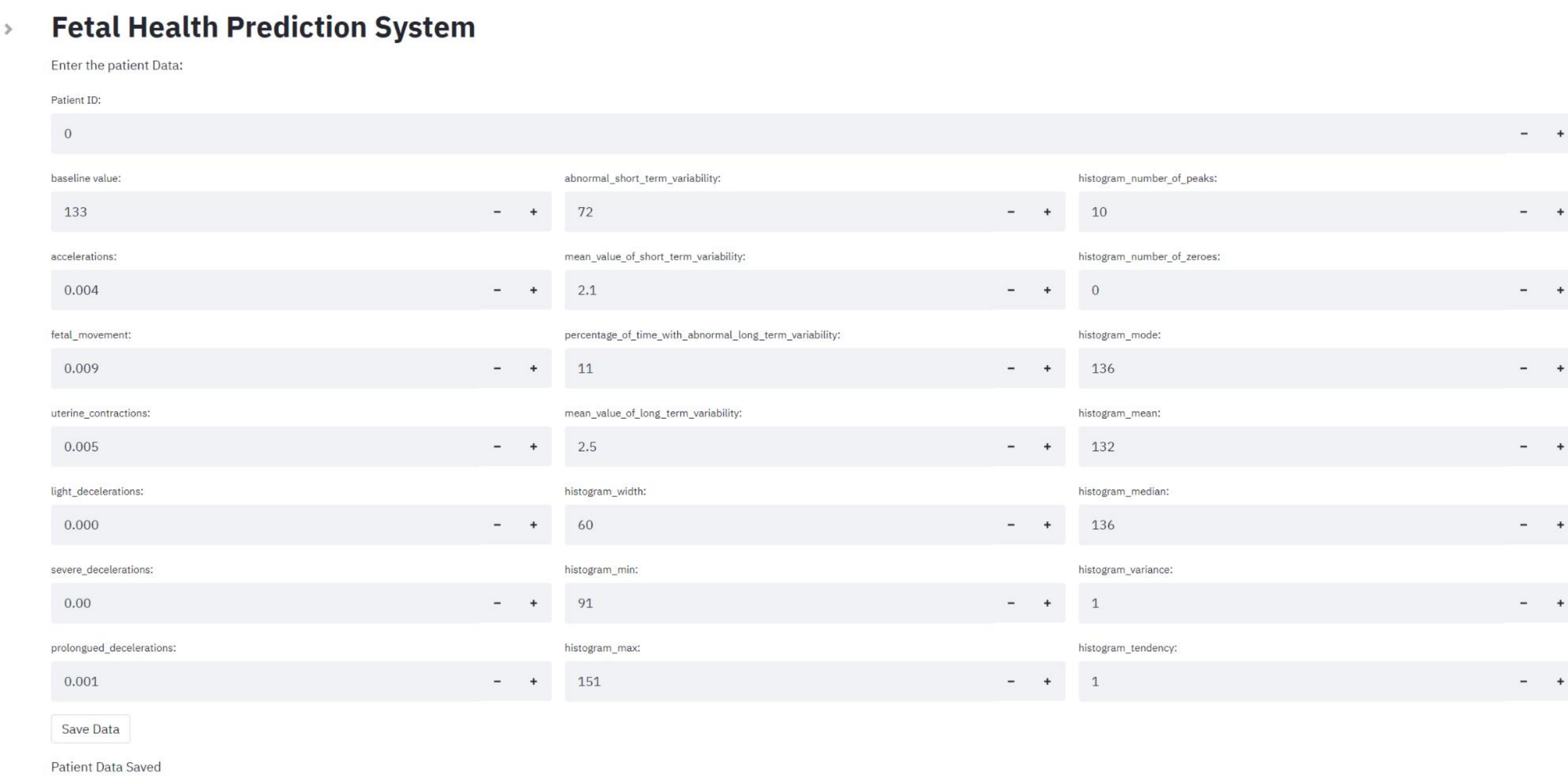


Figure 20: The Data Entry View User Interface

User can enter a patient's ID and their data manually to save it.

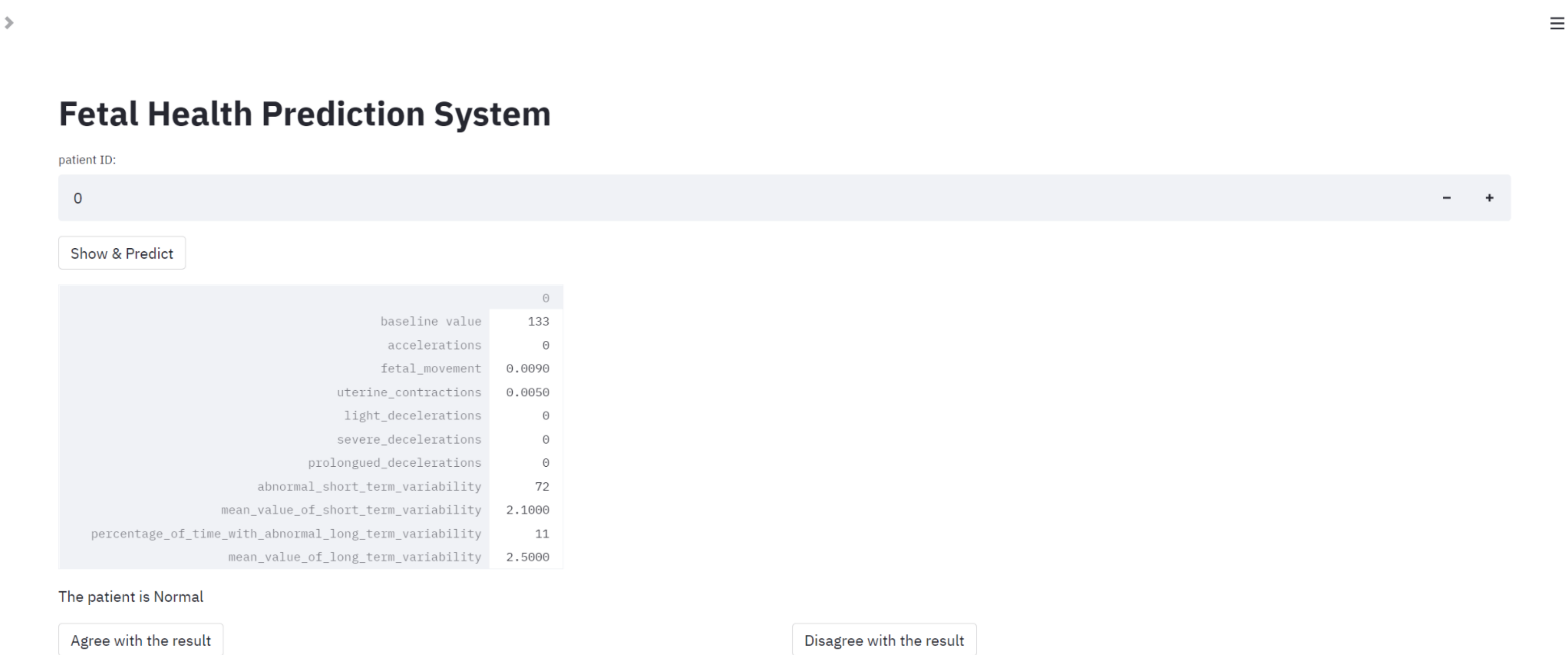


Figure 21: The Doctor's View User Interface

A doctor can enter a patient's ID and then receive their data and the ML model prediction. They can then agree or disagree with the prediction.

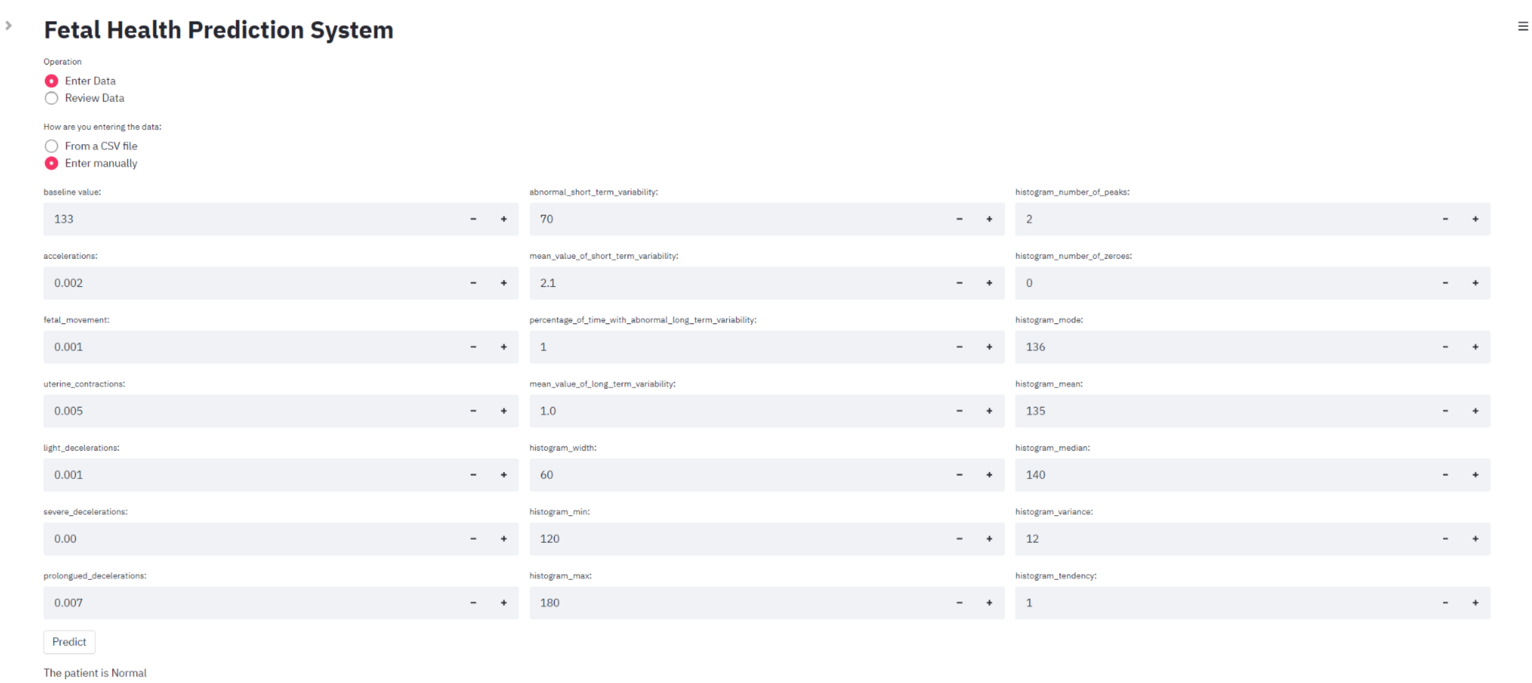


Figure 22: Data Scientist View User Interface 1

Data scientist view when entering data manually. They enter the data manually to get a prediction from the model.

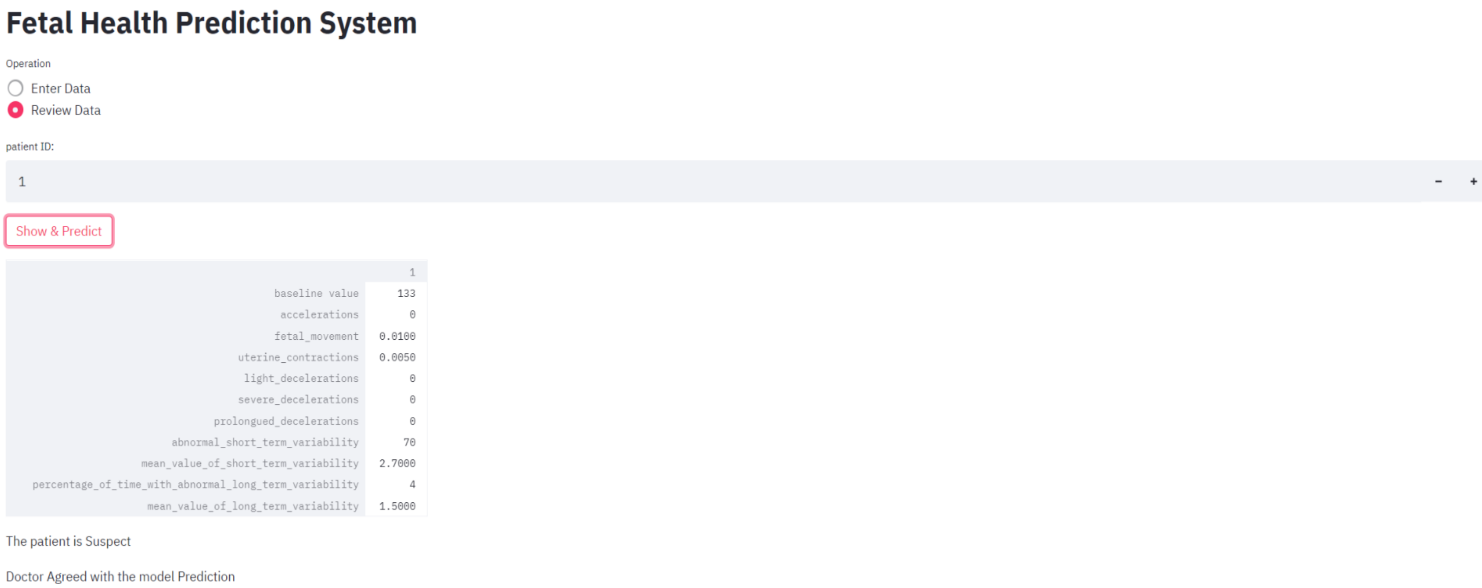


Figure 23: Data Scientist View User Interface 2

Data scientist view when reviewing a patient's data by using their ID and seeing both the ML model prediction and the doctor's evaluation.

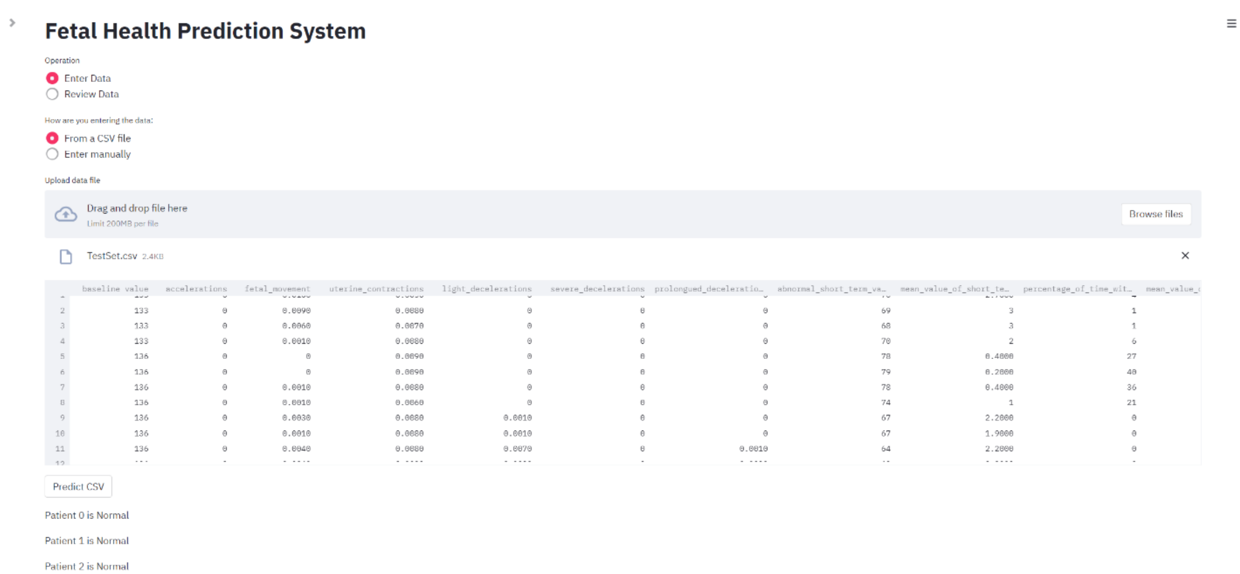


Figure 24: Data Scientist View User Interface 3

Data scientist view when entering data from a csv file. The system shows the CSV data as a table and shows the classifier's predication for each patient.

3.4 Data Analysis & Model Building

This project aims to utilize the data set Fetal Health Classification from Kaggle [5] This dataset contains 2126 records of features extracted from Cardiotocogram exams, which were then classified by three expert obstetricians into 3 classes:

- Normal
- Suspect
- Pathological

The machine learning model will classify each patient as one these classes. This is considered a supervised Multiclass Classification.

In order to build a good model, the data must be studied and analyzed first, starting with understanding the features and their significance

Features

- **'baseline value'** FHR baseline (beats per minute)
- **'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 light decelerations per second
- **'severe_decelerations'** Number of severe decelerations per second
- **'prolongued_decelerations'** Number of prolonged decelerations 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 FHR histogram
- **'histogram_min' Minimum** (low frequency) of FHR histogram
- **'histogram_max' Maximum** (high frequency) of FHR histogram
- **'histogram_number_of_peaks'** Number of histogram peaks
- **'histogram_number_of_zeroes'** Number of histogram zeros
- **'histogram_mode'** Histogram mode
- **'histogram_mean'** Histogram mean
- **'histogram_median'** Histogram median
- **'histogram_variance'** Histogram variance
- **'histogram_tendency'** Histogram tendency

Target

- **'fetal_health'** Tagged as 1 (Normal), 2 (Suspect) and 3 (Pathological)

Next, we need to evaluate the target to find out if the data balanced or not, plotting a pie chart of fetal_health values shows a clear visual of the balance/imbalance of the data.

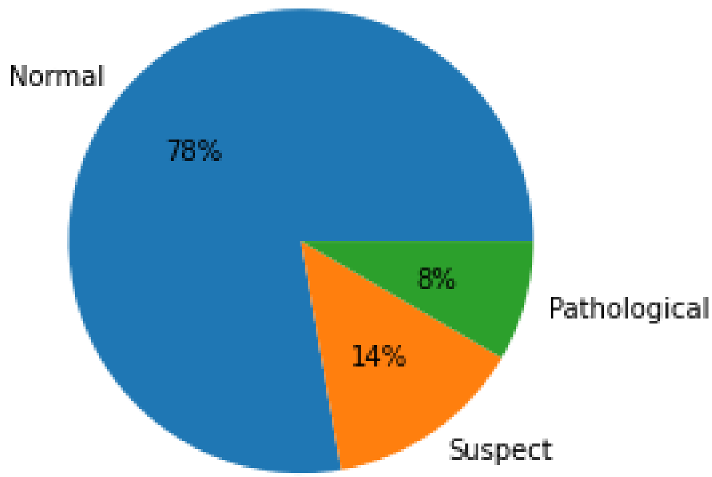


Figure 25: The Distribution of the Target Labels

The pie graph shows that the data is imbalanced, which may provide misleading classification accuracy.

To ensure high and consistent accuracy, multiple accuracy measures are used:

- Confusion Matrix
- Precision
- Recall
- F1 Score

To understand the features and their distribution, histograms are plotted to show their range values.

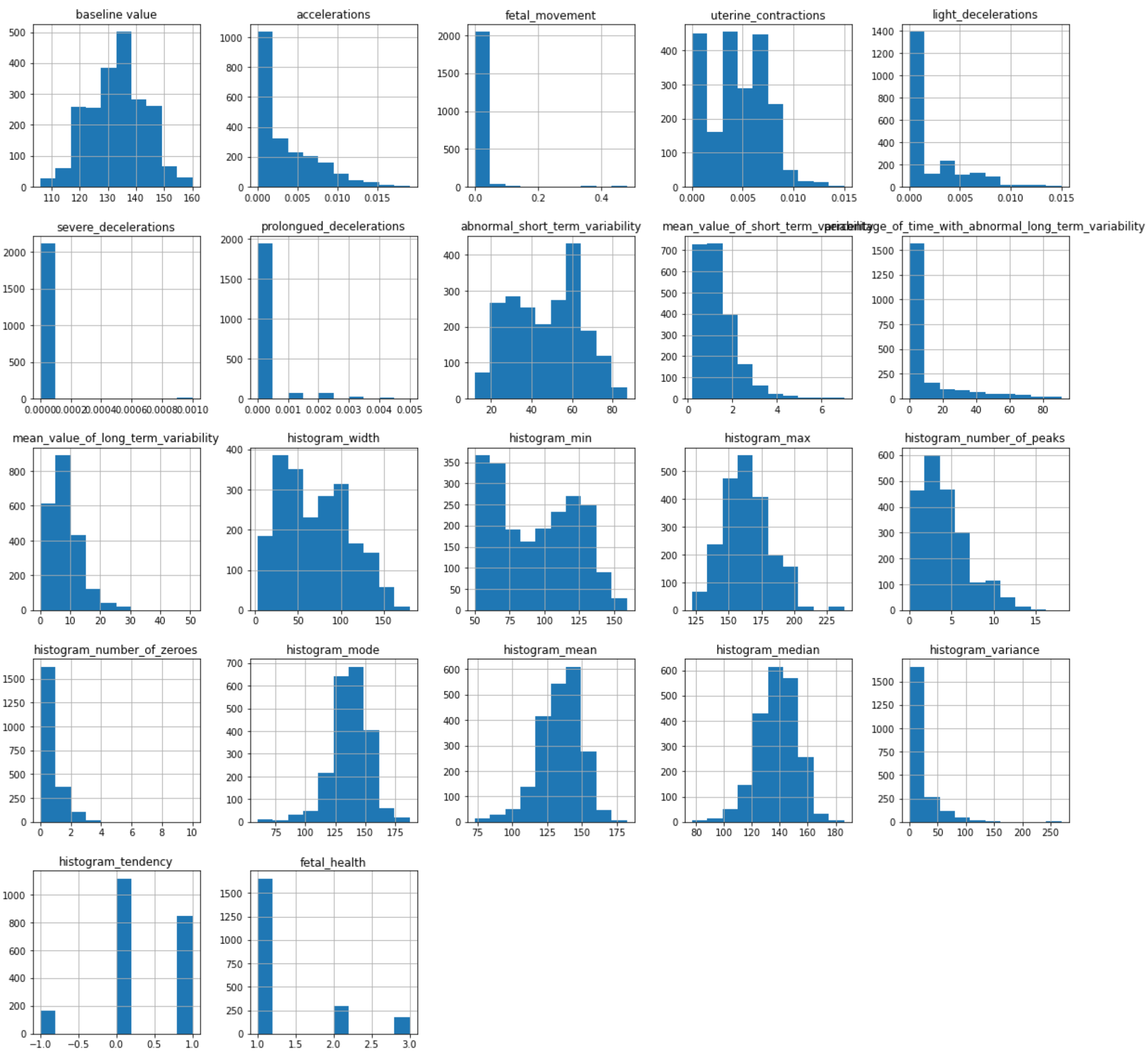


Figure 26: Histograms for Features

To understand how the features are relate to the target a heatmap map showing the correlation between each feature and all other features is plotted .

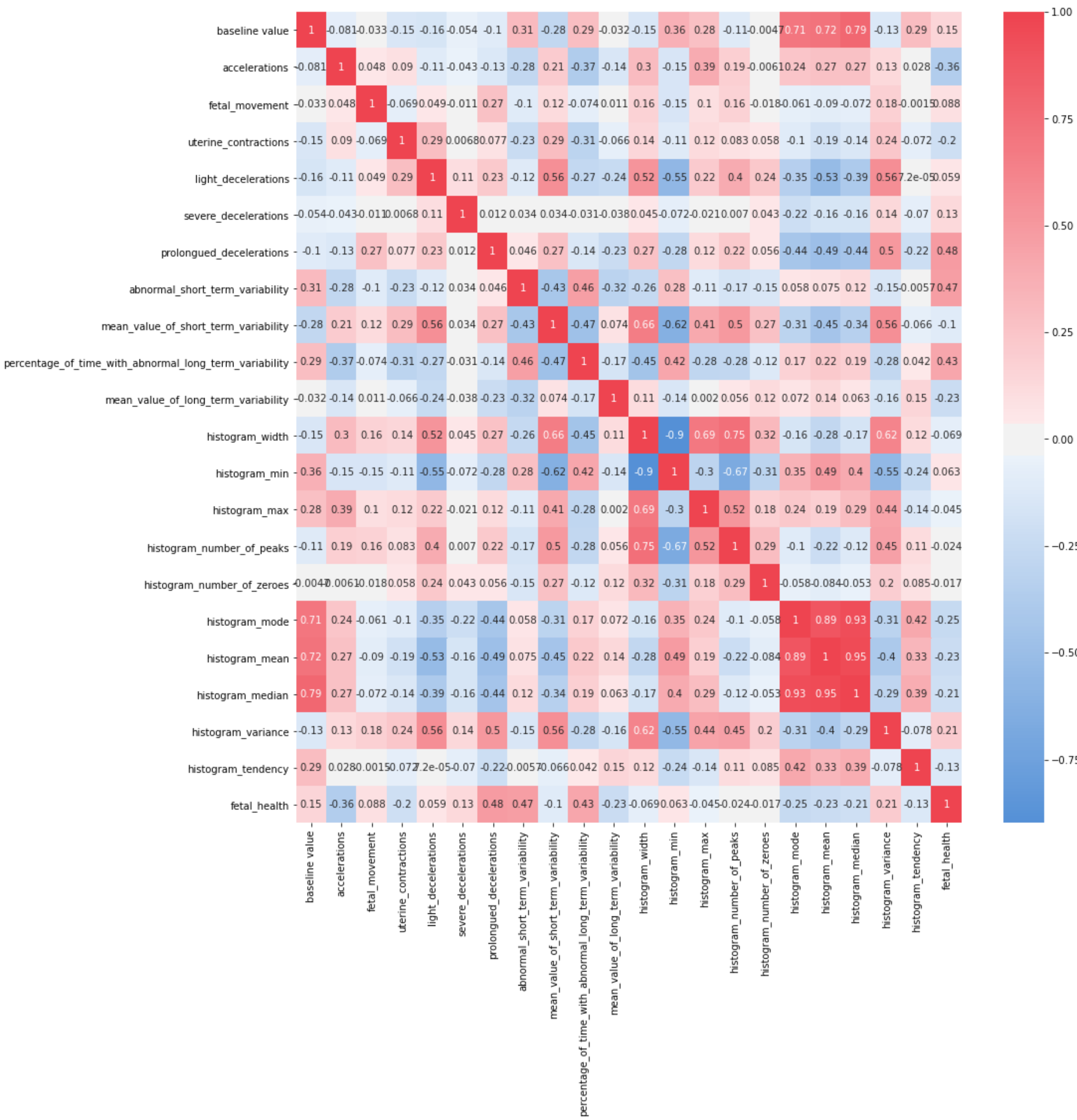


Figure 27: Features Heatmap

Showing the correlation between all features.

Most machine learning algorithm require or benefit from that the features data are in the same range of values, a boxplot is used to show the range of values for each feature.

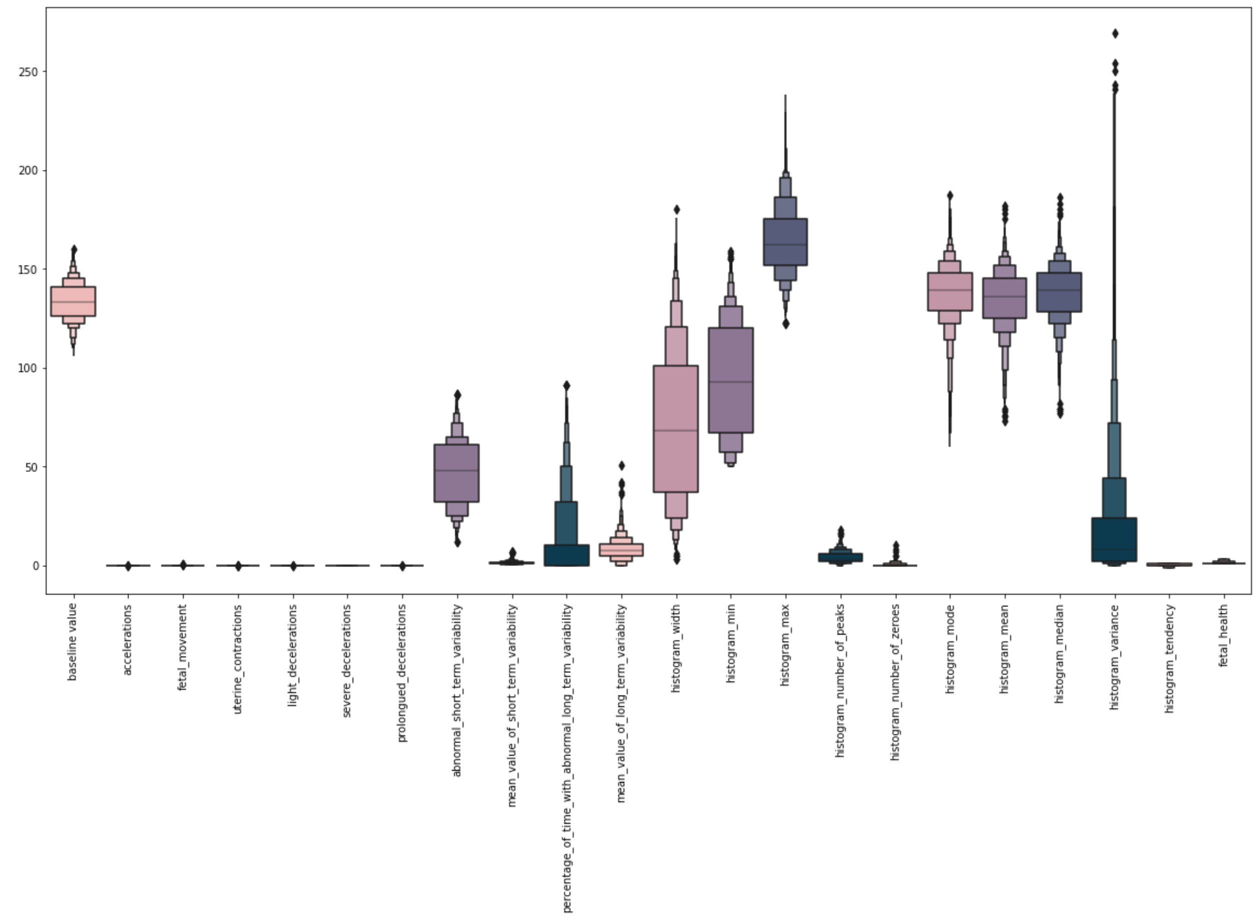


Figure 28: Features Box Plots

Showing value ranges for all features.

Since the feature values have differs greatly between the different features it would be benefitable to Standardize the data, this would the machine learning achieve better accuracy.
It is worth noting that in the feature when using the model the data entering the model need also need to be standardized.

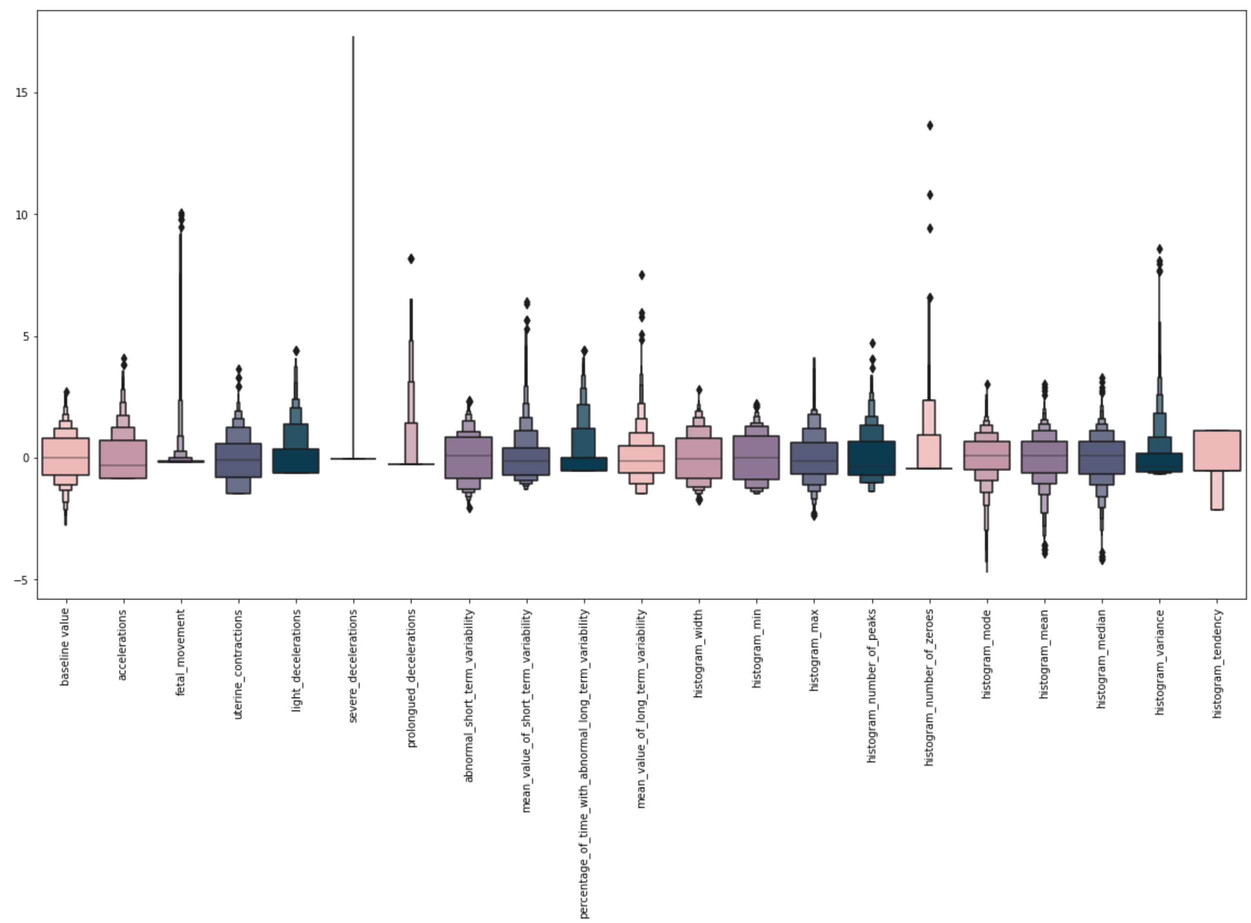


Figure 29: Features Box Plots after Standardization

There are many classification algorithms, such as decision trees, random forest and SVC, many of which will achieved +80% accuracy, but since this is a medical problem the accuracy matters a lot even in few percentages, a way to maximize the accuracy is to combine multiple algorithms in a voting classifier, where each algorithm vote for a target value and classifier choose the class with most votes, this increases the accuracy by around 2%, which is significate in a medical problem. The algorithms used for the voting classifier are the following:

- Decision Tree Classifier
- Random Forest Classifier
- Support Vector Classification

- Gradient Boosting Classifier
- Multi-layer Perceptron Classifier

When using a single classifier, the hyper-parameters used can heavily affect the quality of the classifie r to find the ideal hyper-parameters GridSearchCV is used, which allow for testing the classifier with different combinations of hyper-parameters.

Since we are a voting classifier, it is necessary to find the best hyper-parameters for the classifiers used in the voting classifier.

The following are the hyper-parameters pool used by GridSeachCV.

```
parametersDT = {
    'criterion': ['gini', 'entropy'],
    'max_depth': range(1, 10),
    'min_samples_split': range(2, 10),
    'min_samples_leaf': range(1,5)
}


parametersRF = {
    'n_estimators': [100, 150, 200, 500, 700],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth' : [4,6,8,12,14,16],
    'criterion' :['gini', 'entropy'],
    'n_jobs':[-1,1,None]
}

parametersSVC = [{'kernel': ['rbf'], 'gamma': [1e-3, 1e-4],
                     'C': [1, 10, 100, 1000]},
                    {'kernel': ['linear'], 'C': [1, 10, 100, 1000]}]

parametersGBoost = {
    "loss":["deviance"],
    "learning_rate": [0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2],
    "min_samples_split": np.linspace(0.1, 0.5, 12),
    "min_samples_leaf": np.linspace(0.1, 0.5, 12),
    "max_depth":[3,5,8],
    "max_features":["log2","sqrt"],
    "subsample":[0.5, 0.618, 0.8, 0.85, 0.9, 0.95, 1.0],
    "n_estimators":[10]
    }

parameterMLP = {
    'max_iter': [3000],
    'hidden_layer_sizes': [(50,50,50), (50, 100, 50), (100,)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam'],
    'alpha': [0.0001, 0.05],
    'learning_rate': ['constant','adaptive'],
}
```

Figure 30: Hyper-Parameters Pool for Classifiers Used

After running for while (this is computationally expensive since GridSearchCV will models and evaluate them with each possible combination), we get the following hyper-parameters.

```
Decision Tree best parameters: {'criterion': 'entropy', 'max_depth': 6, 'min_samples_leaf': 2, 'min_samples_split': 3}
Random Forest best parameters: {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'auto', 'n_estimators': 200, 'n_jobs':
1}
SVC best parameters: {'C': 1000, 'gamma': 0.001, 'kernel': 'rbf'}
GBoost best parameters: {'learning_rate': 0.2, 'loss': 'deviance', 'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf':
0.1, 'min_samples_split': 0.17272727272727273, 'n_estimators': 10, 'subsample': 1.0}
MLP best parameters: {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'adaptive',
'max_iter': 3000, 'solver': 'adam'}
```

Figure 31: Ideal Hyper-Parameters for Each Classifiers

With the ideal hyper-parameters, we get a consistent score of ~94% from all 4-evaluation metrics.

```
********* Voting classifer Results *********
Accuracy    :  0.946031746031746
Recall      :  0.946031746031746
Precision   :  0.9446482818479505
F1 Score    :  0.946031746031746
```

Figure 32: The 4-performance Metrics of the Model

A classification report is printed to show the precision, recall, f1-score, and support for each of the possible 3 patient's conditions.

	precision	recall	f1-score	support
Normal	0.96	0.98	0.97	487
Suspect	0.87	0.78	0.82	95
Pathological	0.96	0.96	0.96	48
accuracy			0.95	630
macro avg	0.93	0.90	0.92	630
weighted avg	0.94	0.95	0.94	630

Figure 33: Model Classification Report

A confusion matrix help visualizes the accuracy of the model.

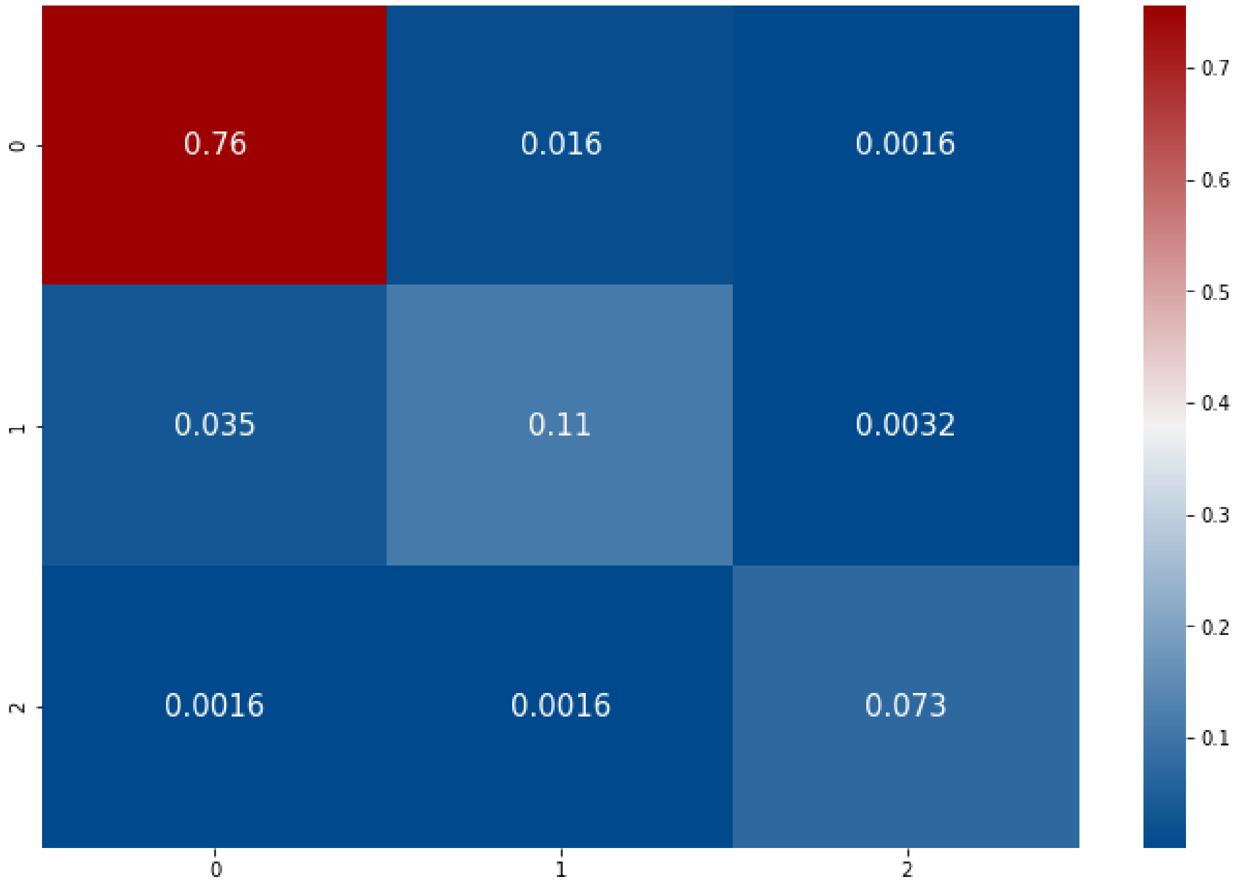


Figure 34: Confusion Matrix Heat Map

3.5 Conclusion and Future Work

The project shows promising results that could benefit the field of gynecology and obstetrics, which in its turn is of added value to the field of medicine - one of the true goals of artificial intelligence. It brings insight to a health professional's decision on the health of a patient's fetus, which is a critical situation considering the issues relating to fetus health are quite complex, and continuous related research is ongoing. The system works with 94% accuracy.

The power of this system lies in its ability to make good predictions with simple use. In fact, actors would have to interact with the system in a minimal amount of time. Data is entered by a data entry person and associated with a patient's ID. The prediction is made and saved, with no human interaction. A healthcare professional's role is to only input the patient's ID.

In addition, the system will maintain its accuracy with evolution, if not increase it. With a data enhancement process service, the system takes in any predicted output made for any user along with the data they inputted, in order to select features and optimize the model. The system allows healthcare providers to give insight on whether they think this prediction is accurate. The intervention of regular humans in the enhancement of the system, will only allow it to perform better over time.

Finally, it is worth looking into how any advancements can be made in the future. To implement a log-in system with different roles assigned for different users, could be of benefit. This is because the system targets only a specific group of people, and is not to be used for personal purposes. A data entry person is in charge of data entry, and a doctor is in charge of seeing their patient's data and the prediction. This is the scope of each actor, and implementing it in a secure system would make the project more efficient. Moreover, connecting to a database with patient's information can be useful for validating patients or keeping hospital records. Last but not least, the data enhancement process -yet to be implemented- can have its own interface, for data scientists to work on. These future developments would make the project a complete package.

To conclude, the project not only shows promising prediction results, but also promising potential. Implementing this project in hospitals and clinics can really shift health professionals' understanding of fetal health and encourage their intervention. This consequently aids in maintaining mothers and fetuses' health, a truly important outcome.

References

[1] Classification Fetal Health

Almishal, Mohammed. “Classification Fetal Health.” Kaggle, Kaggle, 23 Nov. 2020, www.kaggle.com/mohammedmeshil/classification-fetal-health.

[2] 4 Fetal Health Classifiers using GridSearchCV

[3] Comparison of Machine Learning Techniques for Fetal Heart Rate Classification

Cömert, Z., and A.F. Kocamaz. “Comparison of Machine Learning Techniques for Fetal Heart Rate Classification.” Acta Physica Polonica A, vol. 132, no. 3, 2017, pp. 451–454., doi:10.12693/aphyspola.132.451.

[4] Use of Machine Learning Algorithms for Prediction of Fetal Risk using Cardiotocographic Data

Hoodbhoy, Zahra, et al. “Use of Machine Learning Algorithms for Prediction of Fetal Risk Using Cardiotocographic Data.” International Journal of Applied & Basic Medical Research, 11 Oct. 2019, doi:10.5999/aps.2020.01697.s003.

[5] Fetal Health Classification

Larxel. “Fetal Health Classification.” Kaggle, 12 Oct. 2020, www.kaggle.com/andrewmvd/fetal-health-classification.

Woods, Robert. Long-Term Trends in Fetal Mortality: Implications for Developing Countries. 4 Mar. 2011, www.who.int/bulletin/volumes/86/6/07-043471/en/.

Shrivastava, Sajal, et al. Recent Progress, Challenges, and Prospects of Fully Integrated Mobile and Wearable Point-of-Care Testing Systems for Self-Testing. 26 Feb. 2020, pubs.rsc.org/en/content/articlelanding/2020/cs/c9cs00319c#!divAbstract.