FETAL HEALTH PREDICTION USING CARDIOTOCOGRAPHIC DATA

Appagari Bavitha.
(computer science student)

Contents

Introduction

- A major contributor to under-five mortality is the death of children in the 1st month of life.
- The Millennium Development Goal 4, which aimed to reduce underfive mortality by twothirds globally, was
  not able to meet its target.
- Intrapartum complications are one of the major causes of perinatal mortality.
- Fetal cardiotocograph (CTGs) can be used as a monitoring tool to identify highrisk women during labor.

Project objective

- The main objective of our project is to study the precision of machine learning algorithm techniques on CTG data in identifying highrisk fetuses.

- To test larger dataset and to work with real-time systems.

- Lay health-care workers in low and middle-income countries can use this model to triage pregnant women in remote areas for early referral and further management.

Literature Survey

- “Use of Machine Learning Algorithms for Prediction of Fetal Risk using Cardiotocographic Data” from Department of Paediatrics and Child Health, The Aga Khan University, Department of Artificial Intelligence, Ephlux Pvt Ltd., Karachi, Pakistan, Cardiology Care for Children, Pennsylvania, USA.
- Same dataset, CTG data of 2126 pregnant women were obtained from the University of California Irvine Machine Learning Repository has been used.
- Various ML algorithms were trained with dataset and model’s highest precision was 92%.

Abstract

- Fetal cardiotocograph (CTGs) can be used as a monitoring tool to identify highrisk women during labor.

- CTG data of 2126 pregnant women were obtained from the University of California Irvine Machine Learning Repository.

- We are going to train different machine learning classification models using CTG data. Sensitivity, precision, and F1 score for each class and overall accuracy of each model will be obtained to predict normal, suspect, and pathological fetal states.

- The models which we are going to use are K-Nearest Neighbors Classifier, Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, XGB Classifier. Model with best performance on specified metrics will be identified.

- In this project, we will use CTG data of 2126 pregnant women were obtained from the University of California Irvine Machine Learning Repository.
- The dataset, a CSV file contains values of baseline values, accelerations, uterine contractions, light decelerations etc. We have taken the dataset from Kaggle.
- Ten different machine learning classification models were trained using CTG data.
- So we believe that our project will help in classifying fetal health in order to prevent child and maternal mortality.

Dataset

- For dataset design, we, Fetal cardiotocograph (CTGs) can be used as a monitoring tool to identify highrisk women during labor.
- CTG data of 2126 pregnant women were obtained from the University of California Irvine Machine learning Repository.
- The dataset, a CSV file contains values of baseline values, accelerations, uterine contractions, light decelerations etc.
- We have taken the dataset from Kaggle.

Methodologies

- For balancing the dataset we use Synthetic Minority Over-sampling Technique (SMOTE) balancing technique.
- K-Nearest Neighbours Classifier

The number of nearest neighbors to a new unknown variable that has to be predicted or classified is denoted by the symbol ‘K’.

- Decision tree classifier

A decision tree is a class discriminator that recursively partitions the training set until each partition consists entirely or dominantly of examples from one class.

Each non-leaf node of the tree contains a split point that is a test on one or more attributes and determines how the data is partitioned.

- Random forest classifier

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the
predictive accuracy of that dataset.

Methodologies (contd)

- Gradient boosting classifier

  As gradient boosting is one of the boosting algorithms it is used to minimize bias error of the model.

  Gradient boosting algorithm can be used for predicting not only continuous target variable (as Regressor) but also categorical target variable (as a Classifier).
- XGB Classifier

  XGBoost can be used directly for regression predictive modeling.

  XGBoost is an efficient implementation of gradient boosting that can be used for regression predictive modeling.
- Classifiers were compared on the bases of the highest average accuracy across all the folds and sensitivity value of each class
- Various statistical techniques were used to compare the performance of the algorithms. These included precision, sensitivity or recall, F1 score, and overall accuracy

Existing work of project - Disadvantages

Traditional CTG

- **Cardiotocography** (**CTG**) is a technique used to monitor the fetal heartbeat and the uterine contractions during pregnancy and labor.
- The machine used to perform the monitoring is called a cardiotocograph.

Computerized CTG

- In existing work the outliers aren’t looked and removed.
- Therefore the accuracy of model is less.

Advantages of our framework

- A recent Cochrane review by Grivell et al. reported a significant reduction in perinatal mortality with computerized CTG (relative risk: 0.20, 95% confidence interval [CI]: 0.04–0.88) as compared to traditional CTG.

- Our framework used computerized CTG.

- We will remove all the outliers in the data which will boost the accuracy.
- Hence we believe our models accuracy will be even more better.

Requirements

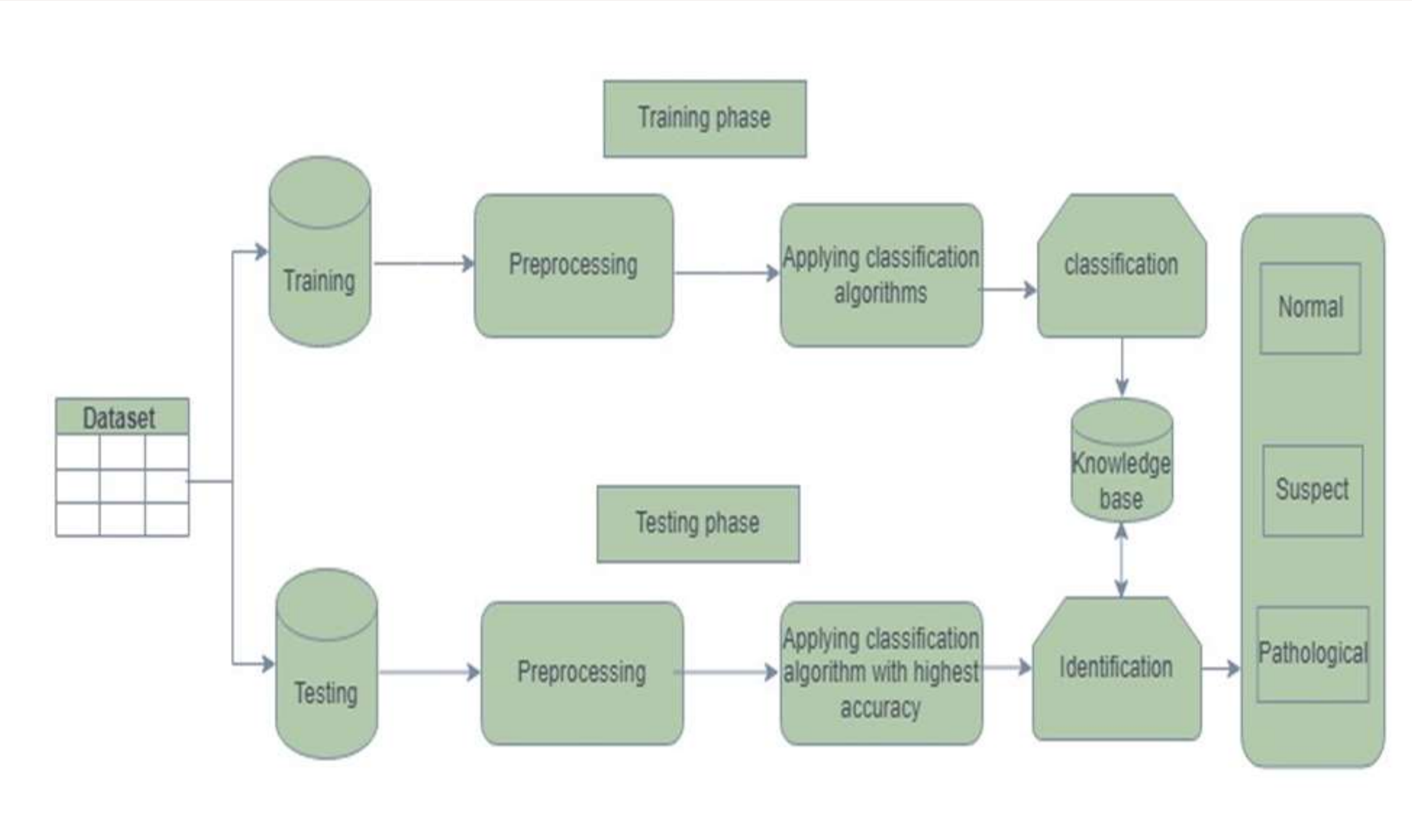
- SOFTWARE REQUIREMENTS
  - ➢ Operating System:- Windows 10
  - ➢ Technology: Python
  - ➢ Software: Anaconda

- HARWARE REQUIREMENTS
  - ➢ RAM:4-GB RAM
  - ➢ PROCESSOR: I5 OR Higher
  - ➢ 64BIT OS

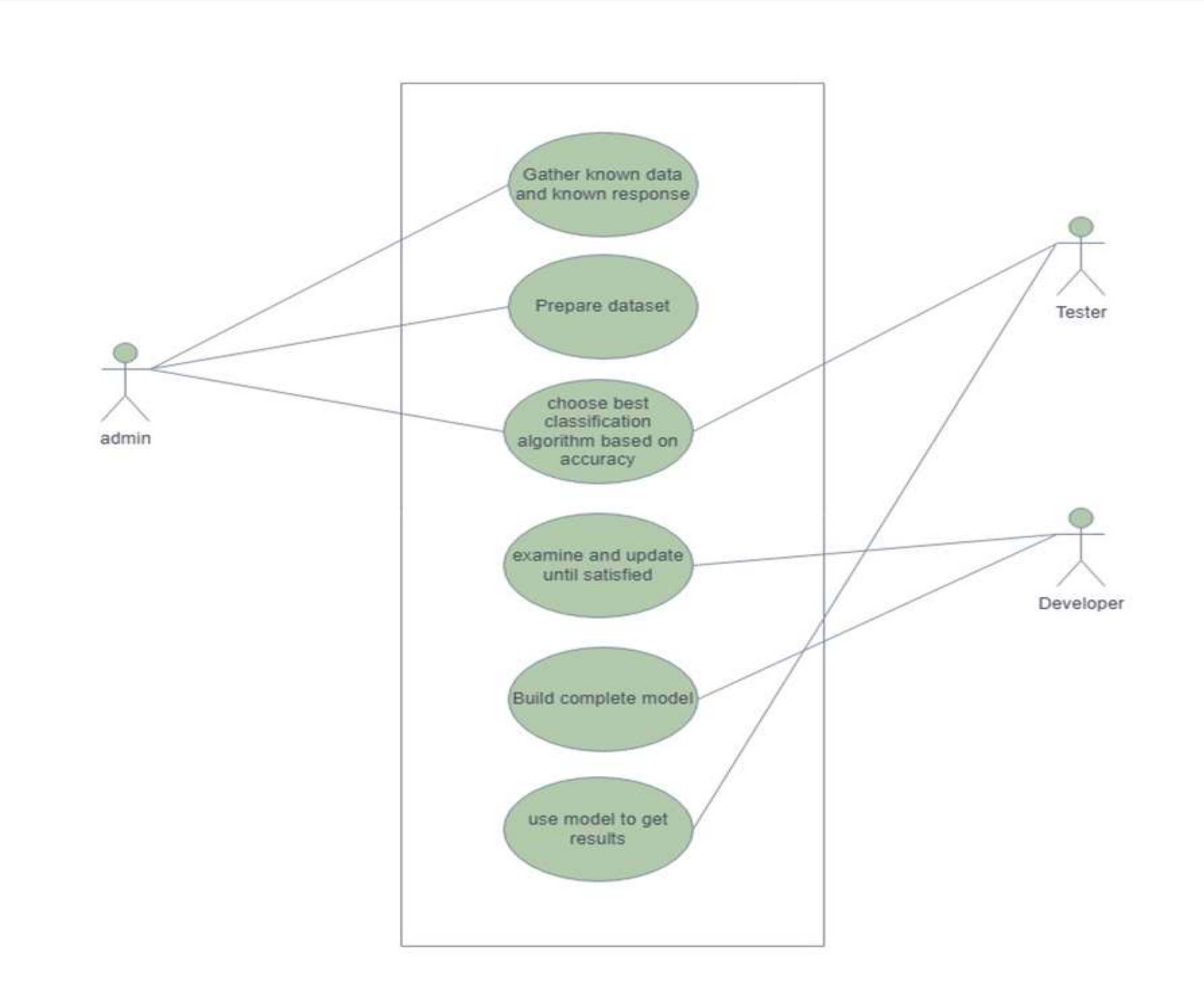
Applications

- We believe this method might be of assistance for health professionals to predict health of fetus.
- This model will help to triage pregnant women in remote areas for early referral and further management.
- This tool can be used to decrease perinatal mortality.

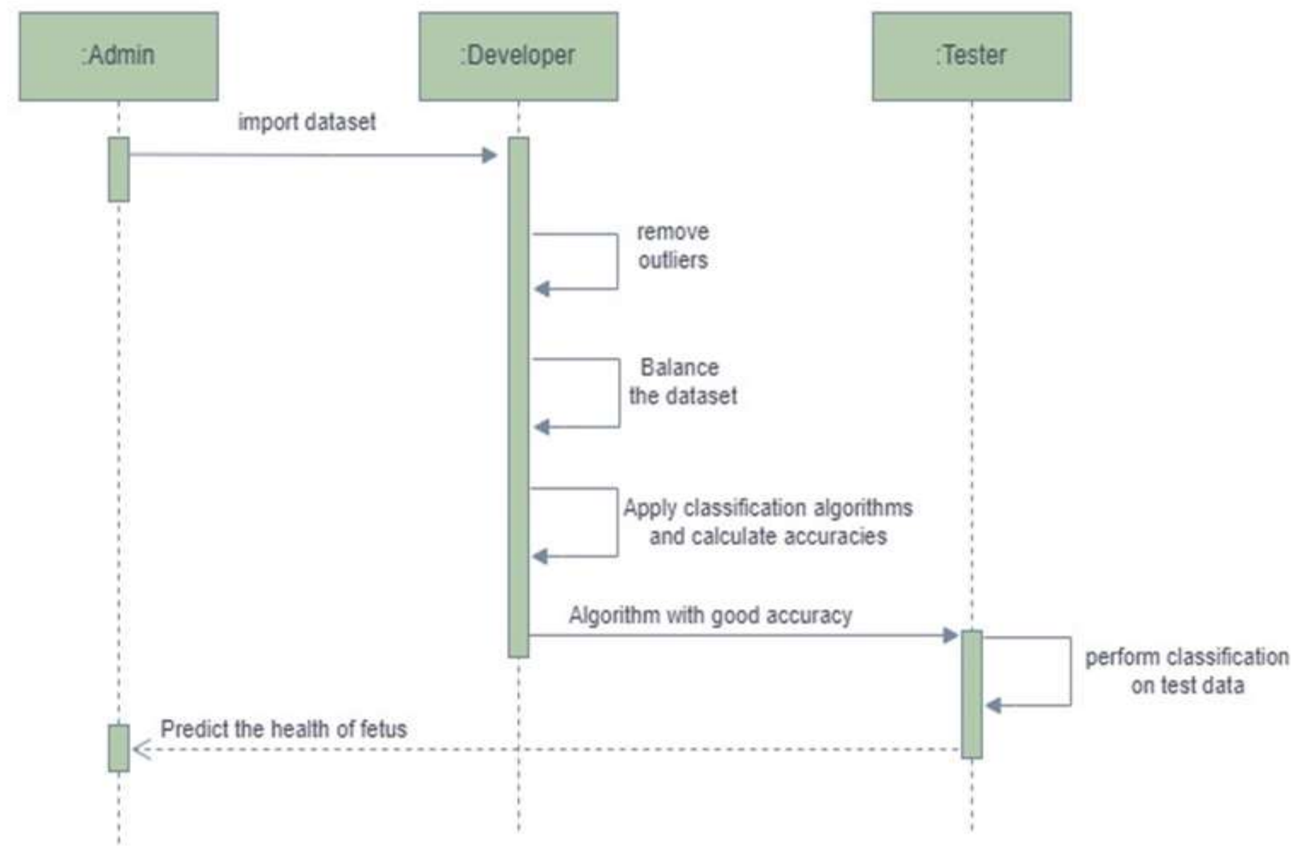
System Architecture



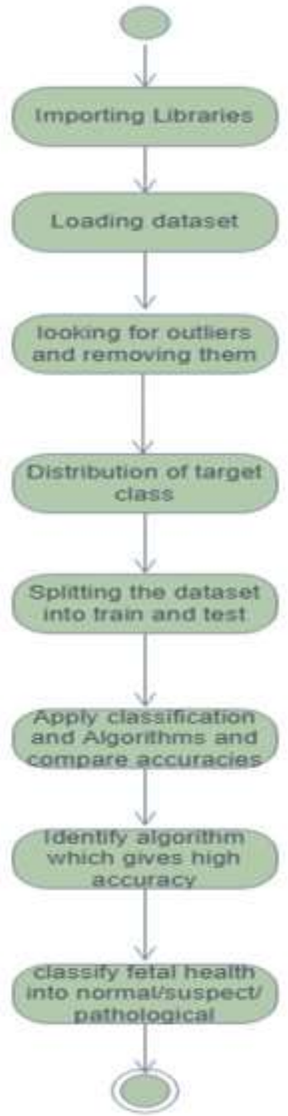
Use Case Diagram



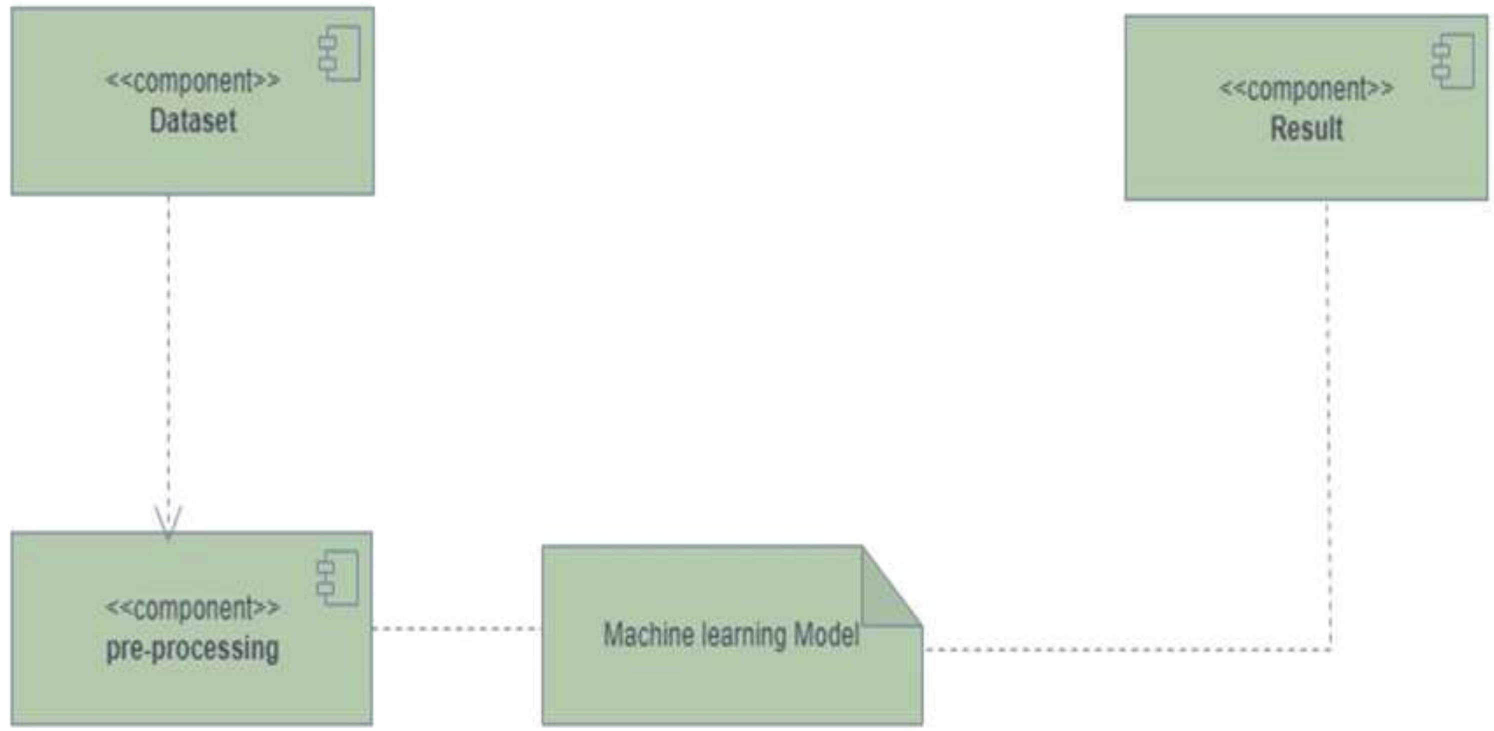
Sequence Diagram



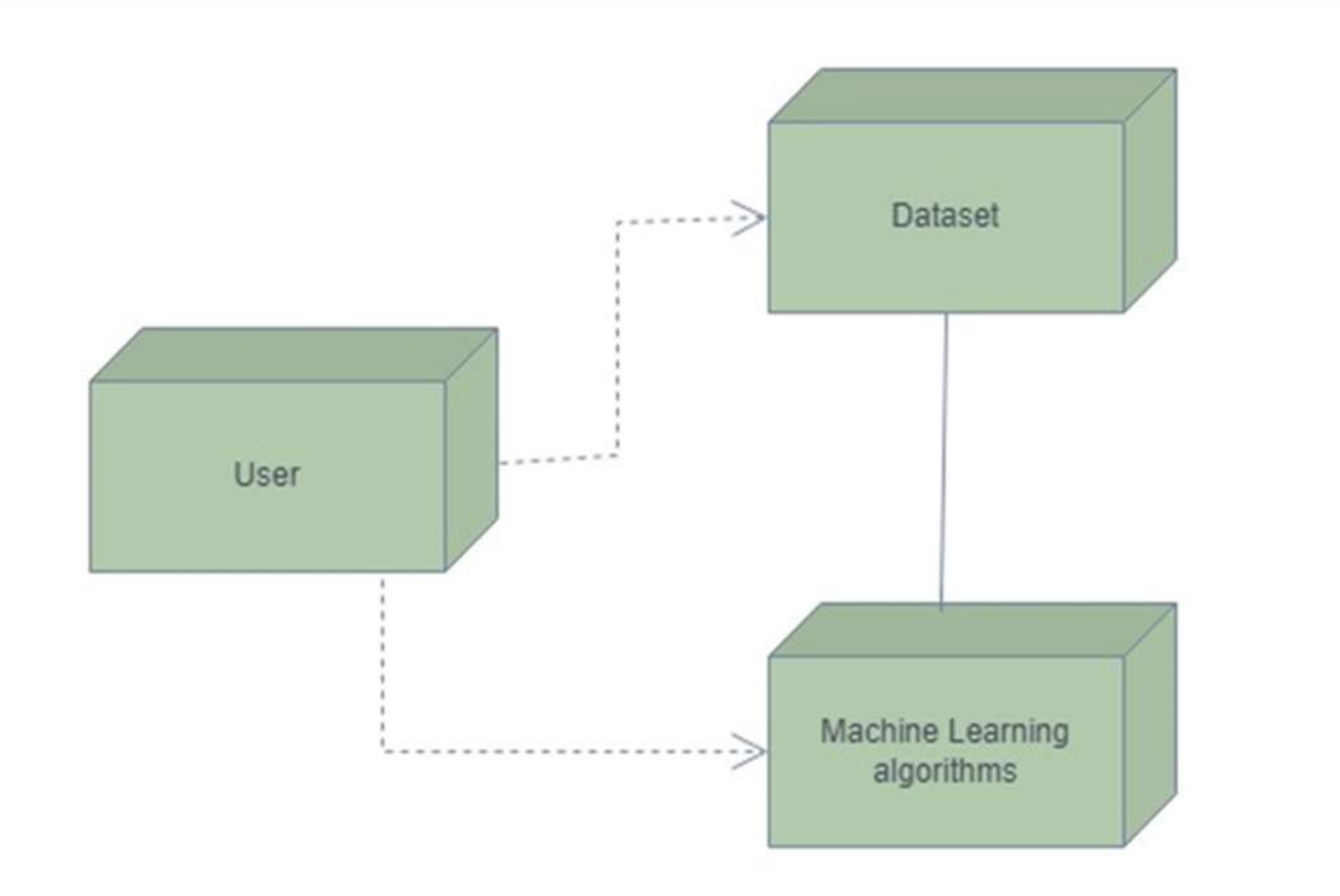
Activity Diagram



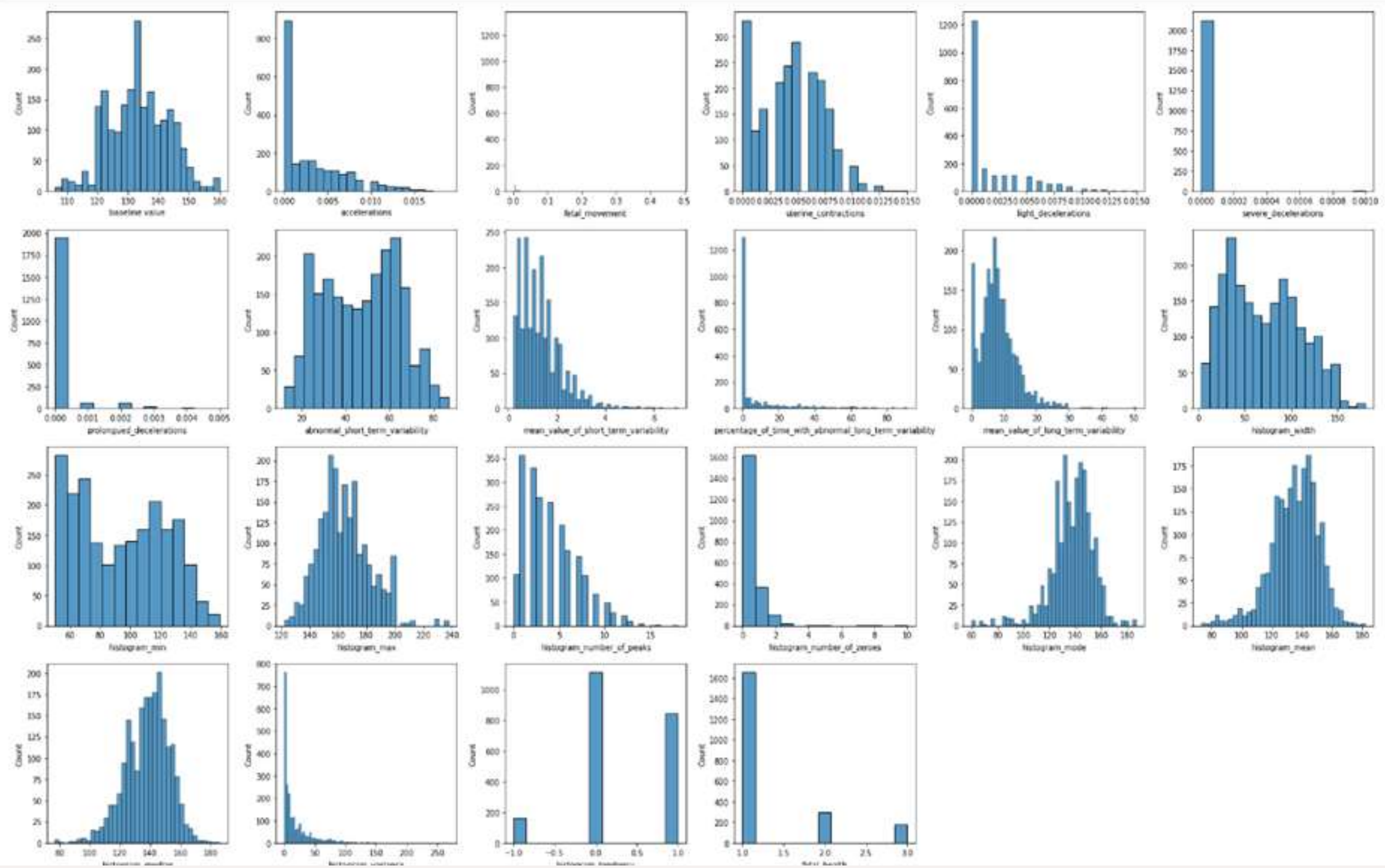
Component Diagram



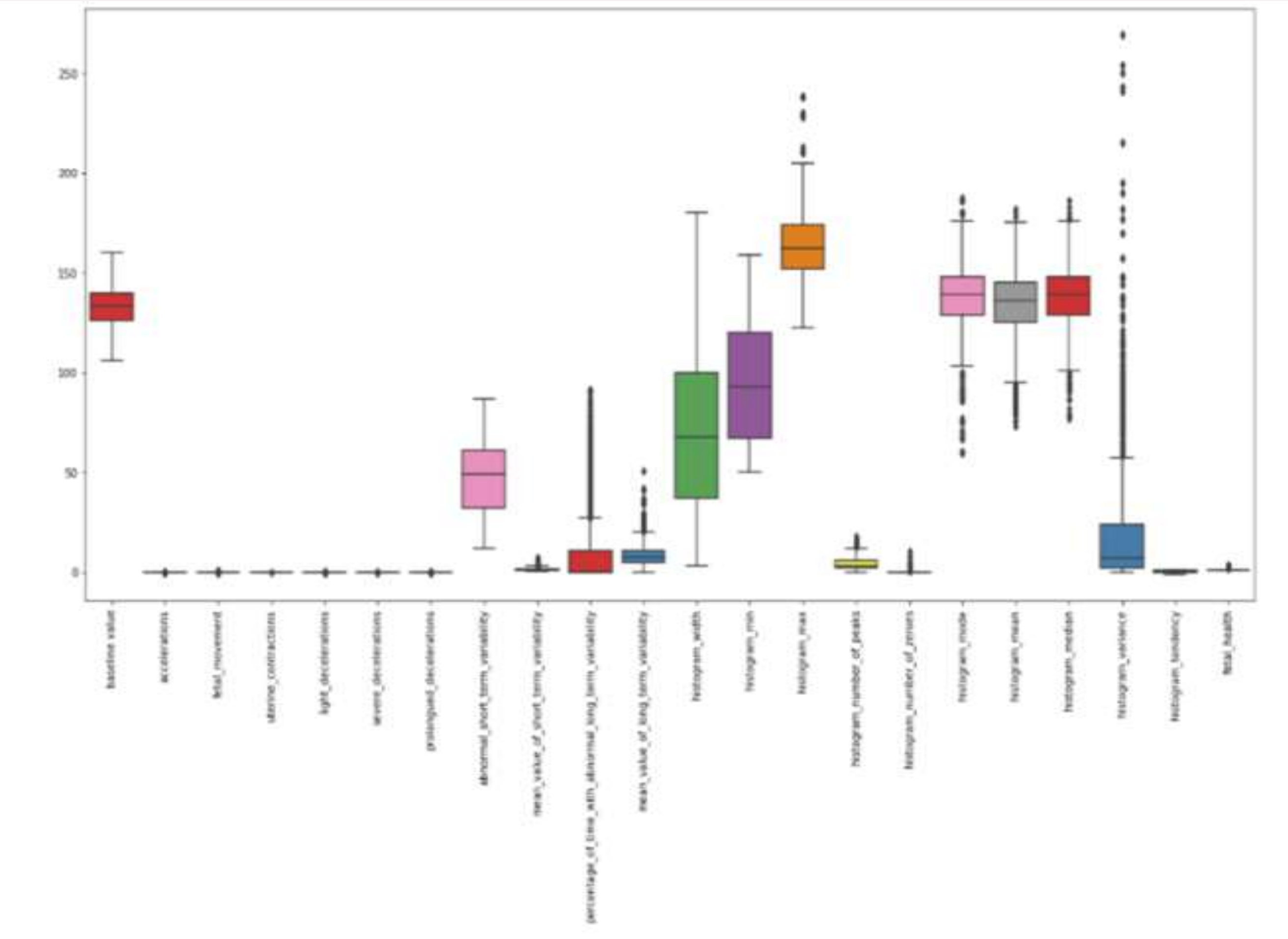
Deployment Diagram



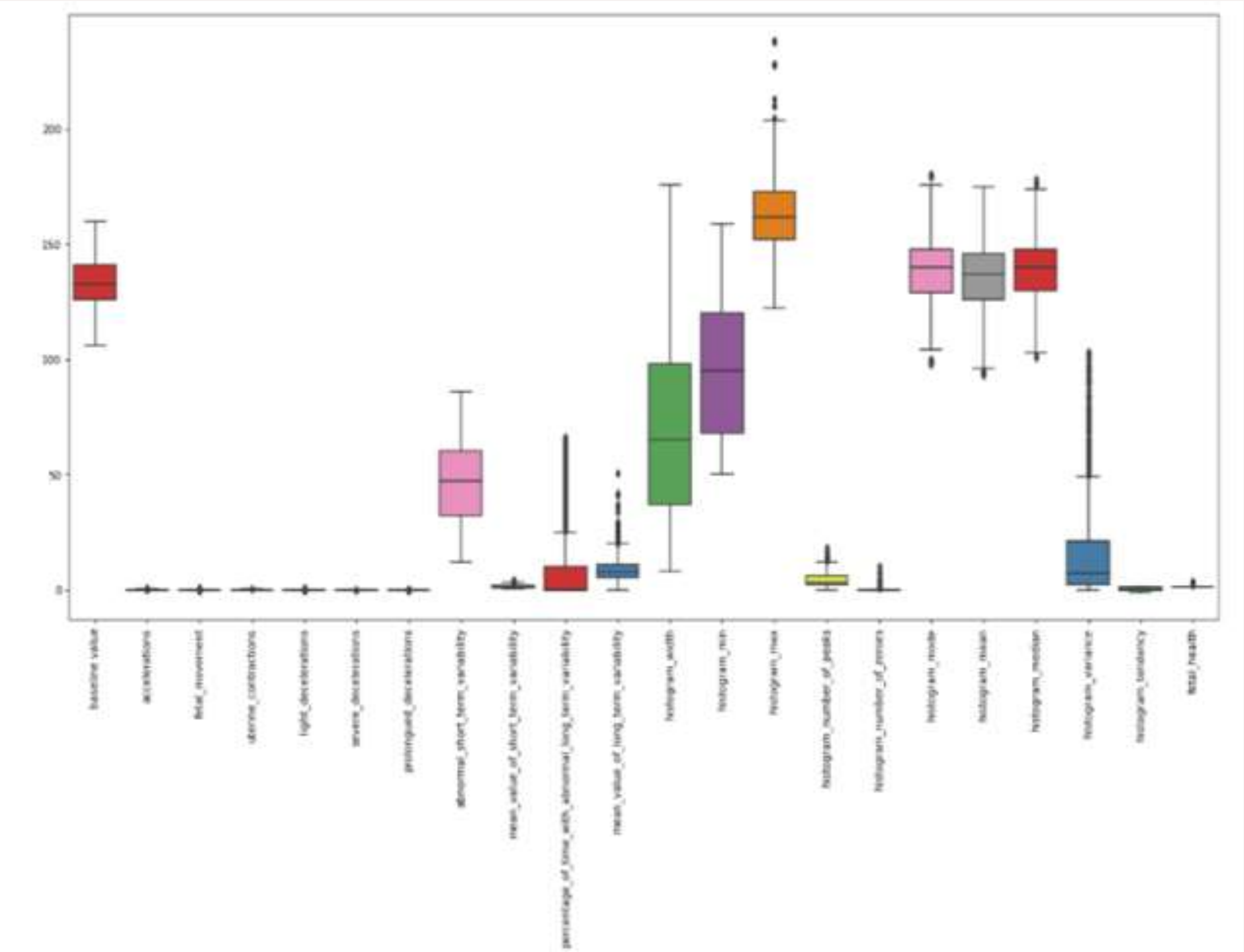
Data Description



Data with outliers



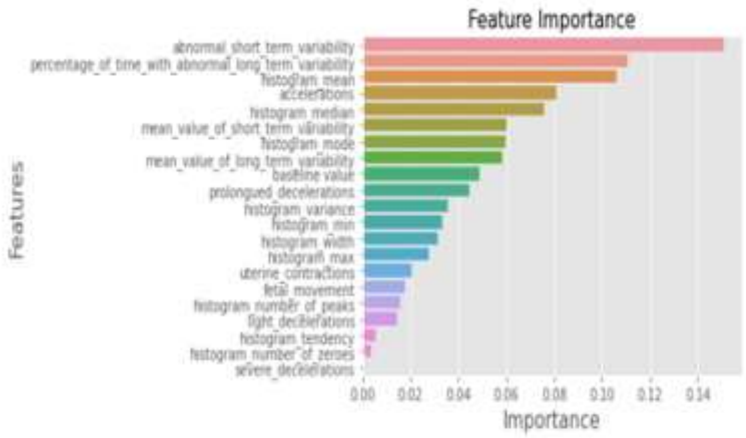
Data without outliers



Accuracies

	Model	Accuracy	Train_acc
4	XGB	0.986831	0.999725
2	RFC	0.984362	0.999725
1	DT	0.977778	0.999725
3	GBC	0.970370	0.986271
0	KNN	0.954733	0.971993

Feature Importance



	Features	Importance
7	abnormal_short_term_variability	0.150614
9	percentage_of_time_with_abnormal_long_term_var...	0.110753
17	histogram_mean	0.105848
1	accelerations	0.081092
18	histogram_median	0.075699
8	mean_value_of_short_term_variability	0.059926
16	histogram_mode	0.059809
10	mean_value_of_long_term_variability	0.058432
0	baseline value	0.048850
6	prolongued_decelerations	0.044428
19	histogram_variance	0.035589
12	histogram_min	0.032986
11	histogram_width	0.031421
13	histogram_max	0.027568
3	uterine_contractions	0.020424
2	fetal_movement	0.017565
14	histogram_number_of_peaks	0.015867
4	light_decelerations	0.014085
20	histogram_tendency	0.005543
15	histogram_number_of_zeroes	0.003325
5	severe_decelerations	0.000175

Output

```
In [28]: input = np.array([[134.0,   0.003,  0.0,    0.008,  0.003,  0.0,    0.0,    16.0,   2.4,    0.0,    23.0,   117.0,  53.0,   170.0
         z=rf_classifier.predict(input)
         print(z)

         for i in range(len(z)):
           if z[i]== 1:
             print("Normal")
           elif z[i]==2:
             print("Suspect")
           else:
             print("Pathological")

[2.]
Suspect
```

```
In [43]: input = np.array([[5.0,24.0,0.0,7.0,0.0,0.0,0.0,0.0,0.0,0.0,4.0,0.0,0.0,12.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0]])
         a=rf_classifier.predict(input)
         print(a)

         for i in range(len(a)):
           if a[i]== 1:
             print("Normal")
           elif a[i]==2:
             print("Suspect")
           else:
             print("Pathological")

[1.]
Normal
```

THANK YOU