

FETALAI:USING MACHINE LEARNING TO PREDICT AND MONITOR FETAL HEALTH

Final Project Report

1. Introduction
 - 1.1. Project overviews
 - 1.2. Objectives
2. Project Initialization and Planning Phase
 - 2.1. Define Problem Statement
 - 2.2. Project Proposal (Proposed Solution)
 - 2.3. Initial Project Planning
3. Data Collection and Preprocessing Phase
 - 3.1. Data Collection Plan and Raw Data Sources Identified
 - 3.2. Data Quality Report
 - 3.3. Data Exploration and Preprocessing
4. Model Development Phase
 - 4.1. Feature Selection Report
 - 4.2. Model Selection Report
 - 4.3. Initial Model Training Code, Model Validation and Evaluation Report
5. Model Optimization and Tuning Phase
 - 5.1. Hyperparameter Tuning Documentation
 - 5.2. Performance Metrics Comparison Report
 - 5.3. Final Model Selection Justification
6. Results
 - 6.1. Output Screenshots
7. Advantages & Disadvantages
8. Conclusion

9. Future Scope

10. Appendix

10.1. Source Code

10.2. GitHub & Project Demo Link

1.INTRODUCTION

1.1 Project overview

"Fetal mortality is a major, but often overlooked, public health problem."

According to the National Center for Health Statistics, about 1 million fetal deaths occur in the United States each year, with almost 26,000 being over the age of 20 weeks gestation. After decades of decline, the US fetal mortality rate has plateaued at this shockingly high number. With fetal mortality comes greater risk of adverse maternal health outcomes, as well as maternal mortality risk increase.

Cardiotocograms (CTGs) measure values such as fetal heart rate, fetal movement, and uterine contractions. "CTGs are a simple and cost-accessible option to assess fetal health, allowing healthcare professionals to take action in order to prevent child and maternal mortality."

Using data from actual patients' CTG exams and their accompanying fetal health outcomes assigned by expert obstetricians, I have determined that automated assessment of fetal health is possible using CTG data.

1.2 Objectives

- Develop Predictive Models:
 - Create machine learning models to predict fetal health outcomes, such as fetal distress, growth abnormalities, and preterm birth.
 - Utilize diverse data sources, including ultrasound images, fetal heart rate, maternal health records, and genetic information.
- Real-time Monitoring and Alerts:
 - Implement real-time monitoring systems that continuously analyze fetal health data and provide alerts for any potential issues.
 - Develop algorithms for early detection of complications, ensuring timely interventions.
- Improve Diagnostic Accuracy:
 - Enhance the accuracy of existing diagnostic tools by integrating machine learning techniques.
 - Reduce the rate of false positives and false negatives in fetal health assessments.

2. Project Initialization and Planning Phase

2.1 Define Problem Statement

Problem Statement: "FetalAI: Utilizing machine learning to predict and monitor fetal health, aiming to enhance prenatal care by providing early detection of potential complications and real-time health monitoring for expectant mothers.

2.2 Project Proposal (Proposed solution)

The "Fetal AI" project leverages machine learning to predict and monitor fetal health using a comprehensive dataset including maternal health indicators and real-time monitoring data. This initiative aims to enhance prenatal care, reduce risks, and improve outcomes for mothers and babies. By providing precise and early detection of potential issues, it aims to improve healthcare delivery and patient satisfaction.

2.3 Initial Project Planning

Initial Project Planning for the "Fetal AI: Using Machine Learning to Predict and Monitor Fetal Health" project involves outlining key objectives, defining scope, and identifying stakeholders. It encompasses setting timelines, allocating resources, and determining the overall project strategy. During this phase, the team establishes a clear understanding of the dataset, formulates goals for analysis, and plans the workflow for data processing. Effective initial planning lays the foundation for a systematic and well-executed project, ensuring successful outcomes in prenatal care and fetal health monitoring.

3. Data Collection and Preprocessing Phase

3.1 Data Collection Plan and Raw Data Sources Identified

The Data Collection and Preprocessing Phase for the "Fetal AI" project involves gathering relevant fetal health data from medical records and sensors, ensuring data quality through verification and addressing missing values. Preprocessing tasks include cleaning, encoding, and organizing the dataset for subsequent exploratory analysis and machine

learning model development. This phase ensures a reliable foundation for accurate fetal health predictions.

3.2 Data Quality Report

The dataset for "Fetal AI: Predicting and Monitoring Fetal Health" is sourced from medical records and sensor data. It includes maternal health indicators and fetal monitoring metrics. Data quality is ensured through thorough verification, addressing missing values, and adhering to ethical guidelines, establishing a reliable foundation for predictive modeling.

3.3 Data Exploration and preprocessing

Data Exploration in "Fetal AI: Using Machine Learning to Predict and Monitor Fetal Health" involves analyzing medical records and sensor data to understand patterns, distributions, and anomalies. Preprocessing includes handling missing data, scaling features, and encoding relevant health indicators. These steps are critical for ensuring data quality and reliability in predicting and monitoring fetal health outcomes.

4. Model Development Phase

4.1 Feature Selection Report

The Feature Selection Report in "Fetal AI" outlines the rationale for selecting maternal health indicators and fetal monitoring metrics (e.g., maternal age, medical history). It evaluates their relevance and importance for accurate predictive modeling in fetal health monitoring and care.

4.2 Model Selection Report

The Model Selection Report for "Fetal AI" details the rationale for choosing machine learning models like Random Forest, Decision Tree, KNN, and Logistic Regression based on their capabilities in handling complex fetal health data, interpretability, and predictive performance, ensuring alignment with project objectives.

4.3 Initial Model Training Code, Model Validation and Evaluation Report

The Initial Model Training Code applies selected algorithms to fetal health data, establishing the foundation for predictive modeling. The subsequent Model Validation and Evaluation Report rigorously assesses model performance using metrics like sensitivity and specificity to ensure reliability in predicting fetal health outcomes.

5. Model Optimization and Tuning Phase

Final Model Selection Justification

- The Random Forest model is the final model chosen because of its best overall performance compared to the other models.
- It captures the variance in the data very well with minimal prediction error.

6. RESULTS

6.1 Output Screenshots

HOME PAGE



PREDICTION PAGE

Fetal Checkup
127.0.0.1:5000/predict

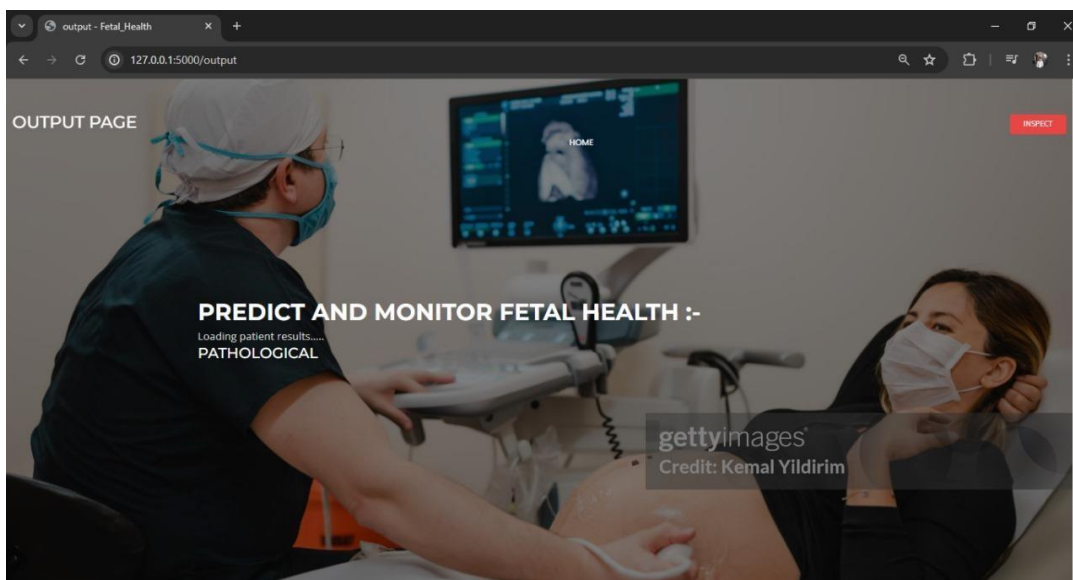
Fetal Health CheckUp

FETAL HEALTH

Baseline Value: 144	Accelerations: 55	Fetal Movement: 55
Uterine Contractions: 55	Light Decelerations: 55	Severe Decelerations: 55
Prolonged Decelerations: 55	Abnormal Short Term Variability: 55	Mean Value of Short Term Variability: 55
Percentage of Time with Abnormal Long Term Variability: 55	Mean Value of Long Term Variability: 55	Histogram Width: 55
Histogram Min: 78	Histogram Max: 175	Histogram Number of Peaks: 55
Histogram Number of Zeroes: 155	Histogram Mode: 155	Histogram Mean: 155
Histogram Median: 55	Histogram Variance: 77	Histogram Tendency: 77

Submit

OUTPUT PAGE



7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1.Improved Early Detection:

- **Timely Identification of Anomalies:** Machine learning algorithms can identify fetal anomalies and potential health issues earlier than traditional methods, allowing for timely interventions.
- **Enhanced Diagnostic Accuracy:** Advanced image analysis and predictive modeling increase the accuracy of detecting conditions like congenital anomalies and growth restrictions.

2. Personalized Care:

- **Tailored Risk Assessments:** ML models can create personalized risk profiles based on a comprehensive analysis of patient data, leading to customized care plans.
- **Optimized Monitoring Protocols:** Continuous monitoring and realtime data analysis enable healthcare providers to adjust monitoring protocols according to the specific needs of each pregnancy.

3. Efficient Resource Utilization:

- **Targeted Interventions:** By accurately identifying high-risk pregnancies, healthcare resources can be allocated more effectively, reducing unnecessary interventions and focusing on cases that require the most attention.
- **Reduced Healthcare Costs:** Early detection and prevention of complications can lead to lower overall healthcare costs by avoiding expensive treatments and hospitalizations.

4. Data-Driven Decision Making:

- **Informed Clinical Decisions:** ML algorithms provide data-driven insights, supporting healthcare providers in making more informed decisions regarding patient care.

- **Continuous Learning:** The use of ML allows for continuous improvement of models as more data becomes available, enhancing the system's accuracy and reliability over time.

5. Scalability and Accessibility:

- **Wide Implementation:** ML models can be scaled and implemented in various healthcare settings, including low-resource environments, to provide high-quality prenatal care.
- **Remote Monitoring:** Fetal AI systems can facilitate remote monitoring, making prenatal care more accessible to patients in rural or underserved areas.

DISADVANTAGES:

1. Data Quality and Availability:

- **Dependence on High-Quality Data:** The effectiveness of ML models relies on the availability and quality of data. Inconsistent or incomplete data can lead to inaccurate predictions.
- **Data Integration Challenges:** Integrating data from various sources (e.g., EHRs, imaging systems) can be complex and may require significant effort to ensure data compatibility and consistency.

2. Ethical and Privacy Concerns:

- **Data Privacy:** The use of sensitive patient data raises concerns about privacy and data security. Ensuring compliance with data protection regulations is essential.
- **Consent and Transparency:** Patients need to be informed about how their data is being used, and obtaining informed consent is crucial to address ethical concerns.

3. Interpretability and Trust:

- **Black-Box Nature of ML Models:** Many ML models, especially deep learning, can be difficult to interpret, making it challenging for healthcare providers to trust and understand the decision-making process.
- **Need for Explainability:** Ensuring that ML models provide explainable and transparent predictions is important for gaining the trust of healthcare providers and patients.

4. Technical and Operational Challenges:

- **Implementation Costs:** Setting up and maintaining ML systems can be expensive, requiring investment in technology and skilled personnel.
- **Continuous Monitoring and Maintenance:** ML models require continuous monitoring and updates to maintain accuracy and adapt to new data, which can be resource-intensive.

5. Risk of Over-Reliance:

- **Potential for Over-Reliance on Technology:** Healthcare providers may become overly reliant on ML systems, potentially overlooking clinical judgment and expertise.
- **Bias and Fairness:** ML models can inadvertently learn and perpetuate biases present in the training data, leading to unfair or biased predictions that need to be addressed through careful model design and evaluation.

8.CONCLUSION

Fetal AI, by integrating machine learning into prenatal care, offers transformative benefits such as early anomaly detection, personalized care, and continuous monitoring, leading to improved maternal and fetal health outcomes. Despite challenges like data quality, ethical concerns, and technical complexities, the applications of Fetal AI—from real-time heart rate analysis to personalized care plans—underscore its potential to revolutionize prenatal healthcare. By addressing these challenges and ensuring ethical and secure deployment, Fetal AI can significantly enhance the precision, effectiveness, and accessibility of prenatal care, contributing to healthier pregnancies and better health outcomes for mothers and babies.

9.FUTURE SCOPE

Future Scope of the Fetal Health Prediction and Management System:

1. Integration with Advanced Imaging Technologies:

- **Enhanced Ultrasound Analysis:** Utilizing more sophisticated imaging techniques and ML models to improve the accuracy and detail of ultrasound scans, allowing for even earlier and more precise detection of fetal anomalies.
- **MRI and 3D Imaging:** Combining ML with MRI and 3D imaging technologies to provide comprehensive fetal assessments, particularly for complex cases where traditional ultrasound might be insufficient.

2. Development of Predictive Analytics:

- **Longitudinal Health Predictions:** Creating models that can predict long-term health outcomes for both the mother and the child based on prenatal data, allowing for early interventions that can improve lifelong health.

- **Multifactorial Risk Assessment:** Incorporating a wider range of data, including genetic, environmental, and lifestyle factors, to provide more holistic and accurate risk assessments.

3. Personalized and Adaptive Care Plans:

- **Dynamic Monitoring Systems:** Developing adaptive systems that continuously learn from new data to provide real-time adjustments to care plans, ensuring that prenatal care is always tailored to the current needs of the patient.
- **Behavioral and Lifestyle Recommendations:** Using predictive models to offer personalized recommendations on diet, exercise, and other lifestyle factors to optimize pregnancy outcomes.

4. Telemedicine and Remote Monitoring:

- **Wearable Devices:** Integrating wearable health monitors with Fetal AI systems to provide continuous, real-time data on fetal and maternal health, enabling remote monitoring and reducing the need for frequent in-person visits.
- **Telehealth Platforms:** Expanding the use of telehealth platforms to facilitate remote consultations and follow-ups, making prenatal care more accessible, especially in underserved or rural areas.

10. APPENDIX

10.1. Source Code

Code Snippets

```
FetalAlIpynb ☆
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[47] #IMPORTING NECESSARY LIBRARIES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('darkgrid')
import warnings
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.linear_model import LogisticRegression
```

```
[ ] #READ THE DATA SET
df=pd.read_csv("/content/fetal_health.csv")
df.head()
```

	baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	me
0	120.0	0.000	0.0	0.000	0.000	0.0	0.0	73.0	
1	132.0	0.006	0.0	0.006	0.003	0.0	0.0	17.0	
2	133.0	0.003	0.0	0.008	0.003	0.0	0.0	16.0	
3	134.0	0.003	0.0	0.008	0.003	0.0	0.0	16.0	
4	132.0	0.007	0.0	0.008	0.000	0.0	0.0	16.0	

5 rows × 22 columns

```
[49] from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[50] df.tail()
```

	baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	me
2121	140.0	0.000	0.000	0.007	0.0	0.0	0.0	79.0	
2122	140.0	0.001	0.000	0.007	0.0	0.0	0.0	78.0	
2123	140.0	0.001	0.000	0.007	0.0	0.0	0.0	79.0	
2124	140.0	0.001	0.000	0.006	0.0	0.0	0.0	78.0	
2125	142.0	0.002	0.002	0.008	0.0	0.0	0.0	74.0	

5 rows × 22 columns

```
—>>> SANITY CHECK OF DATA (i.e, shape of data set)

[51] df.shape

(2126, 22)

sucessfully fetch data ....!!! ( 2,126 ROWS ) AND (22 COLUMNS)
```

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2126 entries, 0 to 2125
Data columns (total 22 columns):
 #   Column                                     Non-Null Count  Dtype  
---  --
 0   baseline_value                           2126 non-null   float64
 1   accelerations                            2126 non-null   float64
 2   fetal_movement                           2126 non-null   float64
 3   uterine_contractions                     2126 non-null   float64
 4   light_decelerations                     2126 non-null   float64
 5   severe_decelerations                     2126 non-null   float64
 6   prolonged_decelerations                  2126 non-null   float64
 7   abnormal_short_term_variability           2126 non-null   float64
 8   mean_value_of_short_term_variability      2126 non-null   float64
 9   percentage_of_time_with_abnormal_long_term_variability  2126 non-null   float64
10   mean_value_of_long_term_variability       2126 non-null   float64
11   histogram_width                           2126 non-null   float64
12   histogram_min                             2126 non-null   float64
13   histogram_max                             2126 non-null   float64
14   histogram_number_of_peaks                 2126 non-null   float64
15   histogram_number_of_zeroes                2126 non-null   float64
16   histogram_mode                             2126 non-null   float64
17   histogram_mean                             2126 non-null   float64
18   histogram_median                          2126 non-null   float64
19   histogram_variance                         2126 non-null   float64
20   histogram_tendency                        2126 non-null   float64
21   fetal_health                              2126 non-null   float64
```

```
#FINDING THE MISSING VALUES
df.isnull().sum()

baseline_value      0
accelerations       0
fetal_movement      0
uterine_contractions 0
light_decelerations 0
severe_decelerations 0
prolonged_decelerations 0
abnormal_short_term_variability 0
mean_value_of_short_term_variability 0
percentage_of_time_with_abnormal_long_term_variability 0
mean_value_of_long_term_variability 0
histogram_width     0
histogram_min       0
histogram_max       0
histogram_number_of_peaks 0
histogram_number_of_zeroes 0
histogram_mode      0
histogram_mean      0
histogram_median     0
histogram_variance   0
histogram_tendency   0
fetal_health        0
dtype: int64

NO MISSING VALUES IN DATA SET
```

```
# FINDING MISSING VALUES IN PERCENTAGE %
df.isnull().sum()/df.shape[0]*100

baseline_value      0.0
accelerations       0.0
fetal_movement      0.0
uterine_contractions 0.0
light_decelerations 0.0
severe_decelerations 0.0
prolonged_decelerations 0.0
abnormal_short_term_variability 0.0
mean_value_of_short_term_variability 0.0
percentage_of_time_with_abnormal_long_term_variability 0.0
mean_value_of_long_term_variability 0.0
histogram_width     0.0
histogram_min       0.0
histogram_max       0.0
histogram_number_of_peaks 0.0
histogram_number_of_zeroes 0.0
histogram_mode      0.0
histogram_mean      0.0
histogram_median     0.0
histogram_variance   0.0
histogram_tendency   0.0
fetal_health        0.0
dtype: float64
```

```
[55] #CHECKING THE DUPLICATE VALUES
df.duplicated().sum()

13

Oops..! THERE ARE 13 DUPLICATE VALUES EXISTED....

# DESCRIPTIVE FUNCTION
df.describe()


```

	baseline_value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolonged_decelerations	abnormal_short_term_variability
count	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000	2126.000000
mean	133.303857	0.003178	0.009481	0.004366	0.001889	0.000003	0.000159	46.990
std	9.840844	0.003866	0.046666	0.002946	0.002960	0.000057	0.000590	17.192
min	106.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	12.000
25%	126.000000	0.000000	0.000000	0.002000	0.000000	0.000000	0.000000	32.000
50%	133.000000	0.002000	0.000000	0.004000	0.000000	0.000000	0.000000	49.000
75%	140.000000	0.006000	0.003000	0.007000	0.003000	0.000000	0.000000	61.000
max	160.000000	0.016000	0.481000	0.015000	0.015000	0.001000	0.005000	67.000

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
baseline value	2126.0	133.303857	9.840844	106.0	126.000	133.000	140.000	160.000
accelerations	2126.0	0.003178	0.003866	0.0	0.000	0.002	0.006	0.019
fetal_movement	2126.0	0.009481	0.046666	0.0	0.000	0.000	0.003	0.481
uterine_contractions	2126.0	0.004366	0.002946	0.0	0.002	0.004	0.007	0.015
light_decelerations	2126.0	0.001889	0.002980	0.0	0.000	0.000	0.003	0.015
severe_decelerations	2126.0	0.000003	0.000057	0.0	0.000	0.000	0.000	0.001
prolongued_decelerations	2126.0	0.000159	0.000590	0.0	0.000	0.000	0.000	0.005
abnormal_short_term_variability	2126.0	46.990122	17.192814	12.0	32.000	49.000	61.000	87.000
mean_value_of_short_term_variability	2126.0	1.332785	0.883241	0.2	0.700	1.200	1.700	7.000
percentage_of_time_with_abnormal_long_term_variability	2126.0	9.846860	18.396880	0.0	0.000	0.000	11.000	91.000
mean_value_of_long_term_variability	2126.0	8.187629	5.628247	0.0	4.600	7.400	10.800	50.700
histogram_width	2126.0	70.445908	38.955693	3.0	37.000	67.500	100.000	180.000
histogram_min	2126.0	93.579492	29.560212	50.0	67.000	93.000	120.000	159.000
histogram_max	2126.0	164.025400	17.944183	122.0	152.000	162.000	174.000	238.000

```
[58] df['fetal_health'] = df['fetal_health'].astype('int')
```

→ HERE TARGET COLUMN CHANGED DATATYPE AS INT

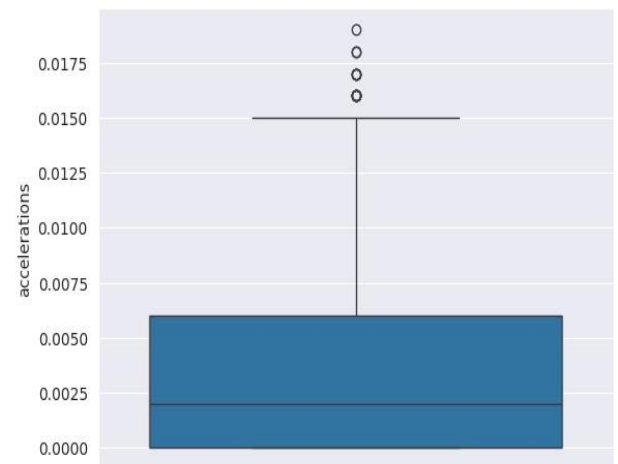
```
[59] df['fetal_health'].value_counts()
```

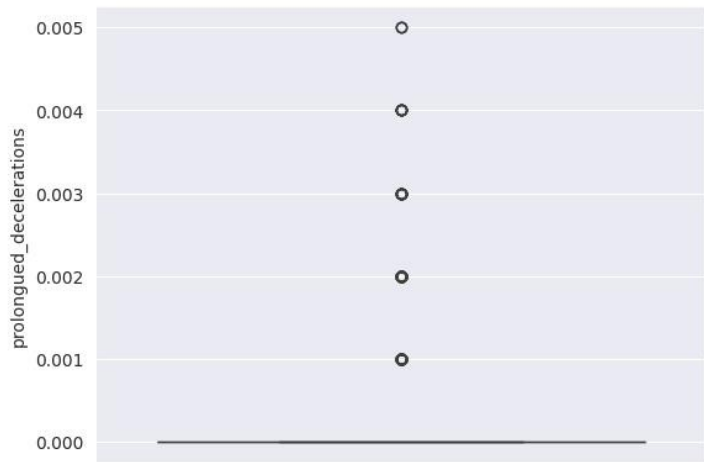
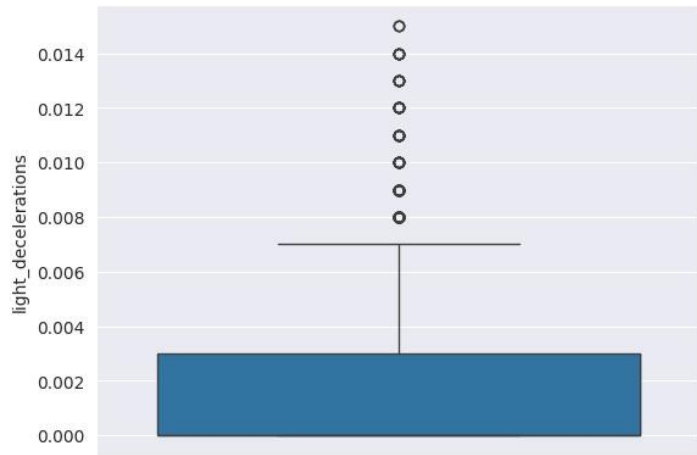
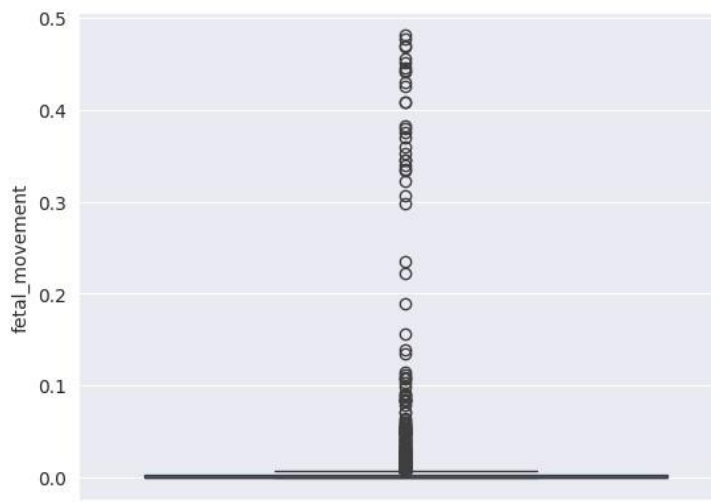
```
fetal_health
1    1655
2     295
3     176
Name: count, dtype: int64
```

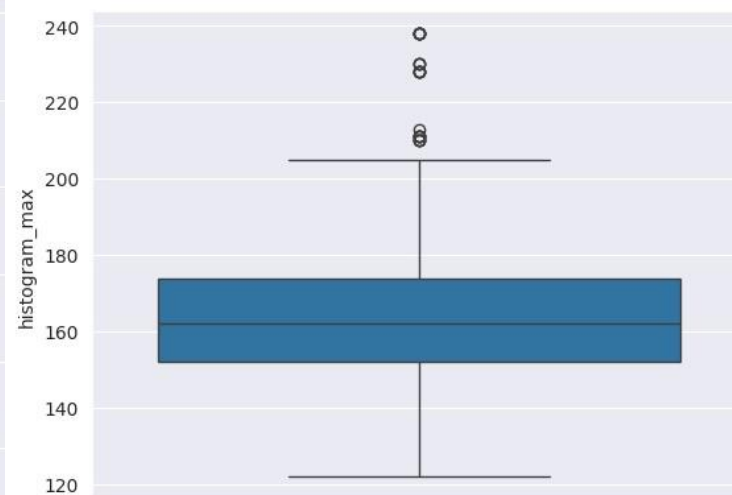
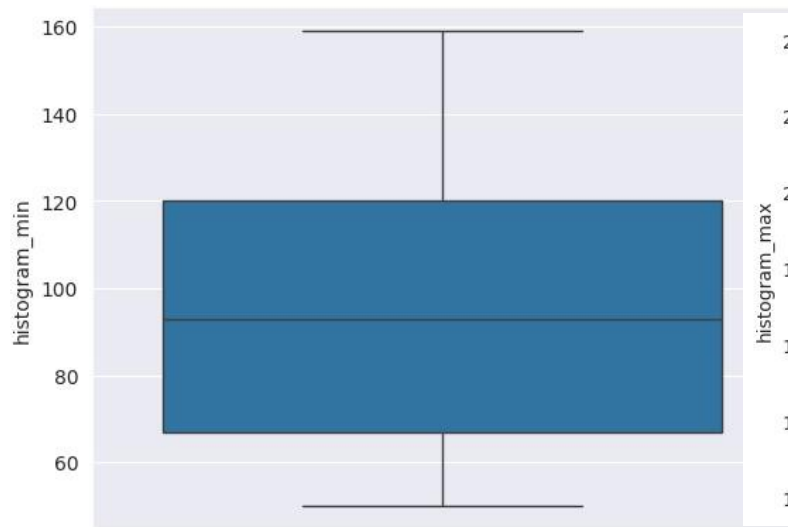
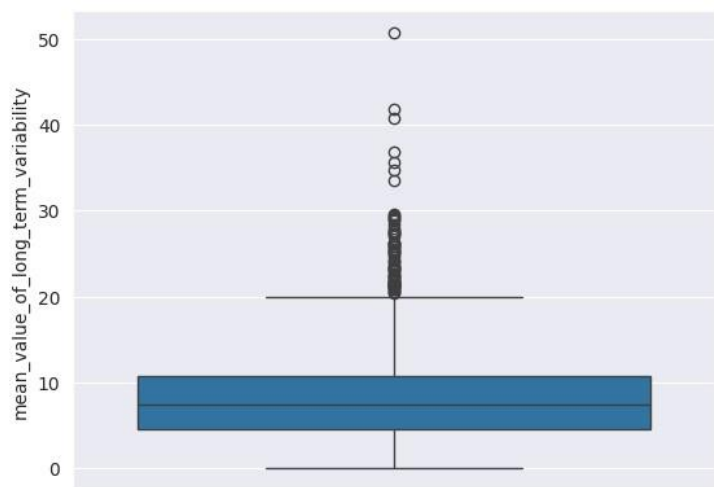
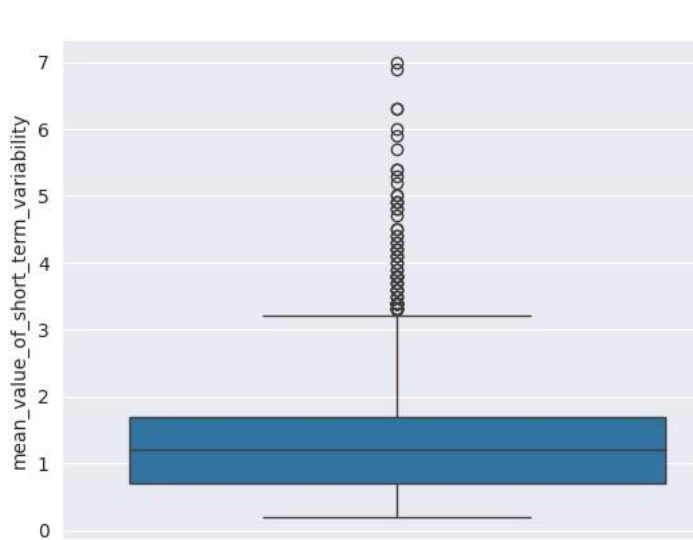
```
# CHECKING COLUMNS
df.columns
```

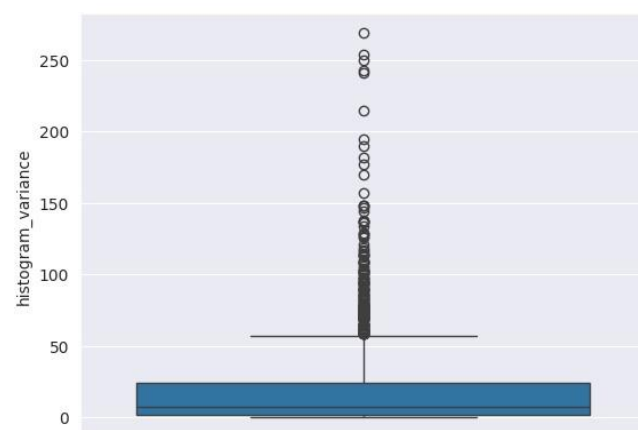
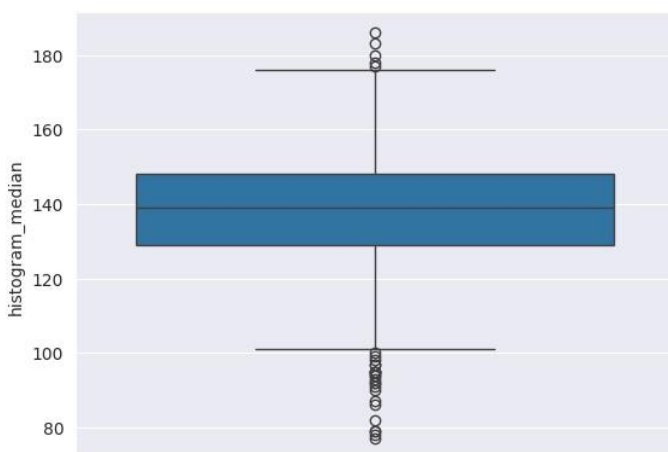
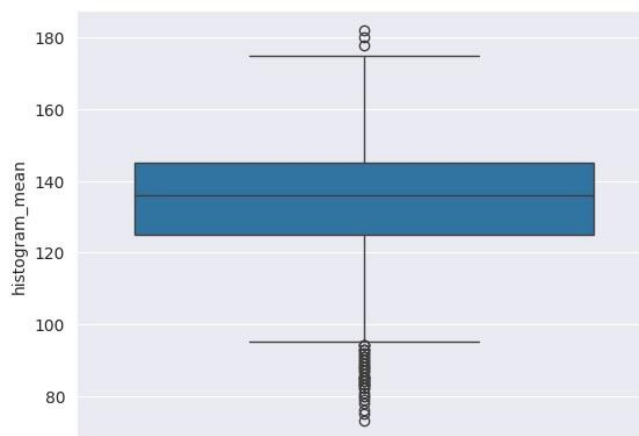
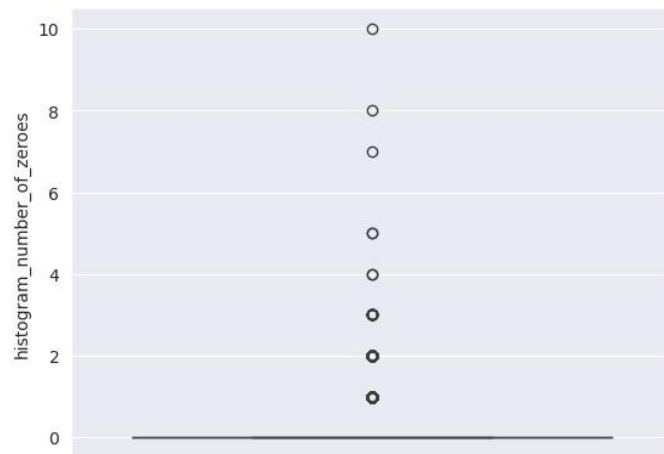
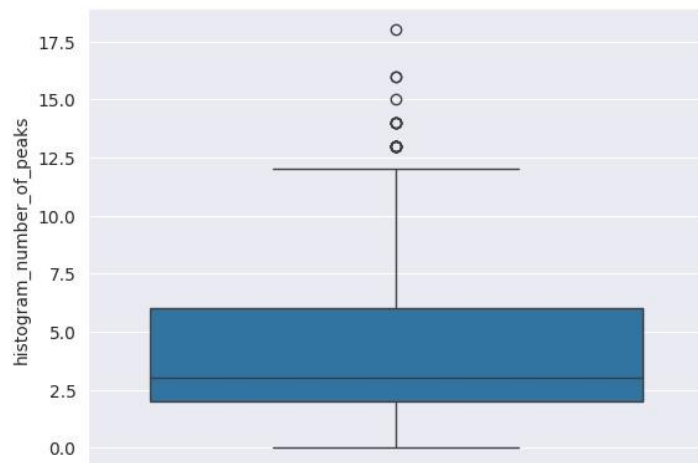
```
Index(['baseline value', 'accelerations', 'fetal_movement',
       'uterine_contractions', 'light_decelerations', 'severe_decelerations',
       'prolongued_decelerations', 'abnormal_short_term_variability',
       'mean_value_of_short_term_variability',
       'percentage_of_time_with_abnormal_long_term_variability',
       'mean_value_of_long_term_variability', 'histogram_width',
       'histogram_min', 'histogram_max', 'histogram_number_of_peaks',
       'histogram_number_of_zeroes', 'histogram_mode', 'histogram_mean',
       'histogram_median', 'histogram_variance', 'histogram_tendency',
       'fetal_health'],
      dtype='object', length=20)
```

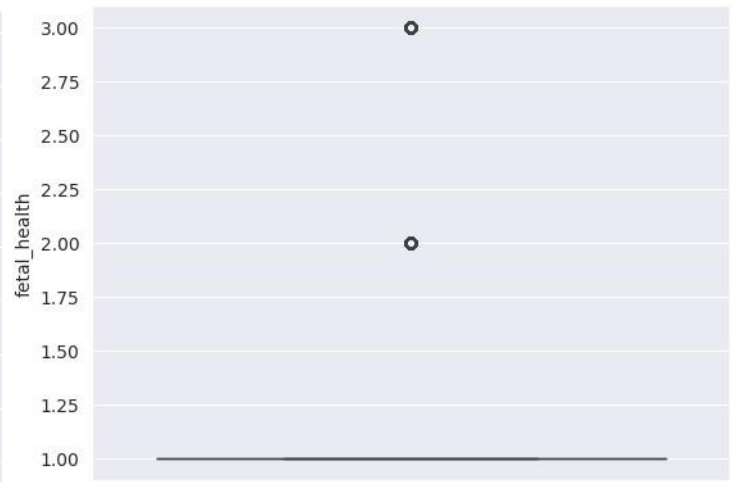
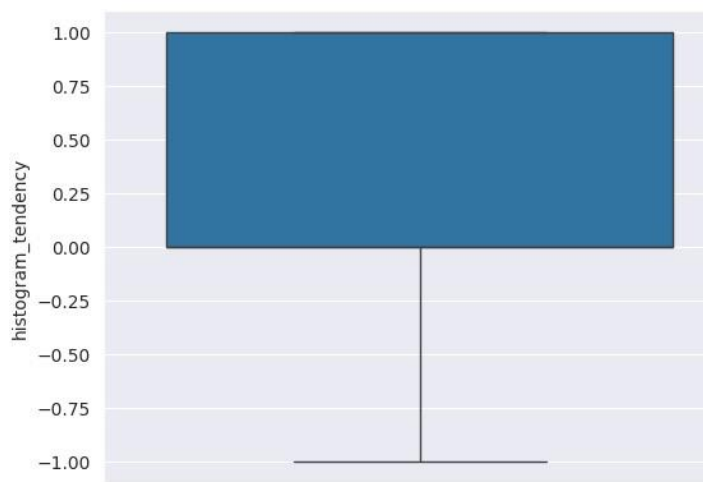
```
for i in df.select_dtypes(include=['int64', 'float64']).columns:
    sns.boxplot(df[i])
    plt.show()
```



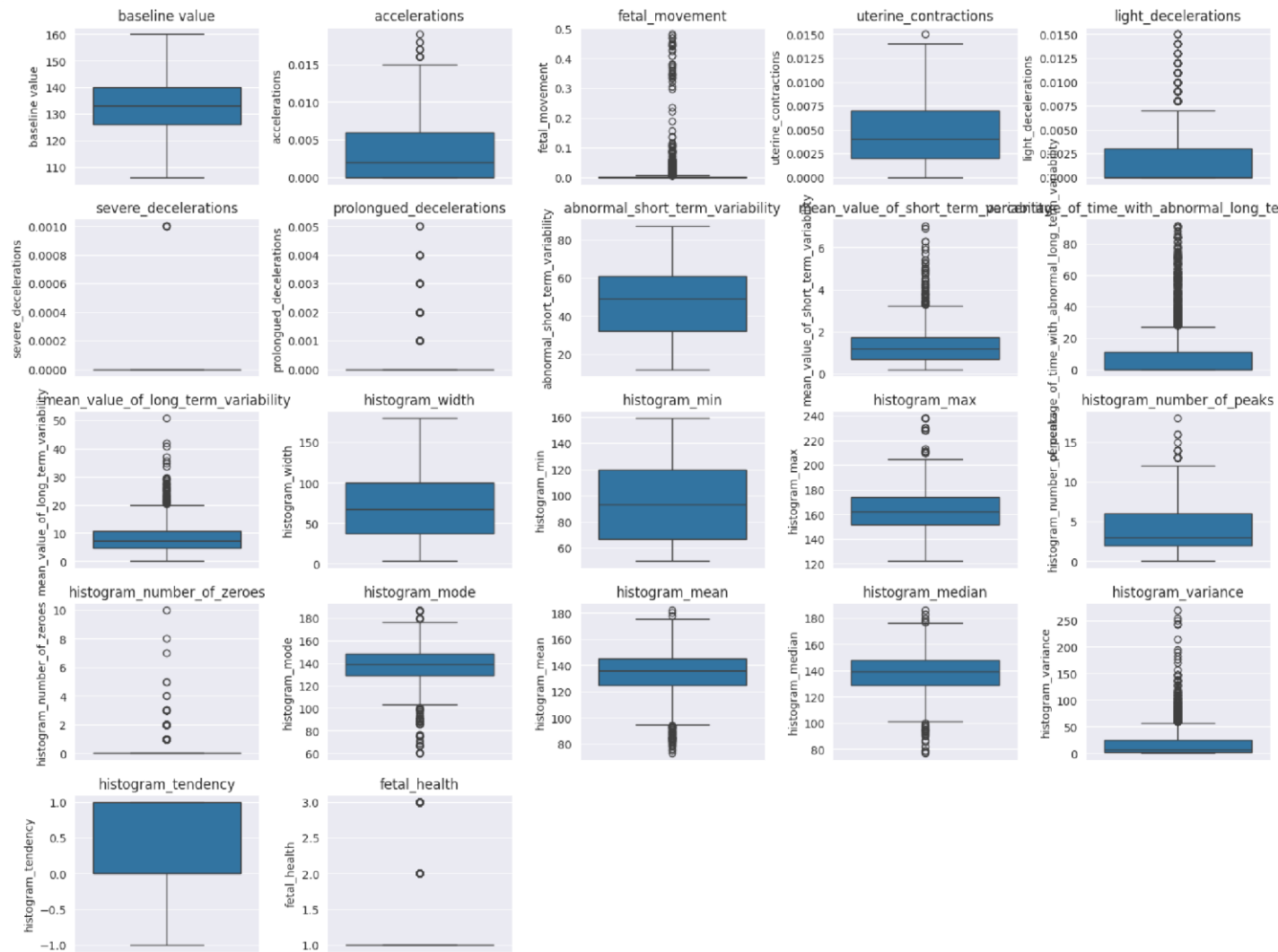








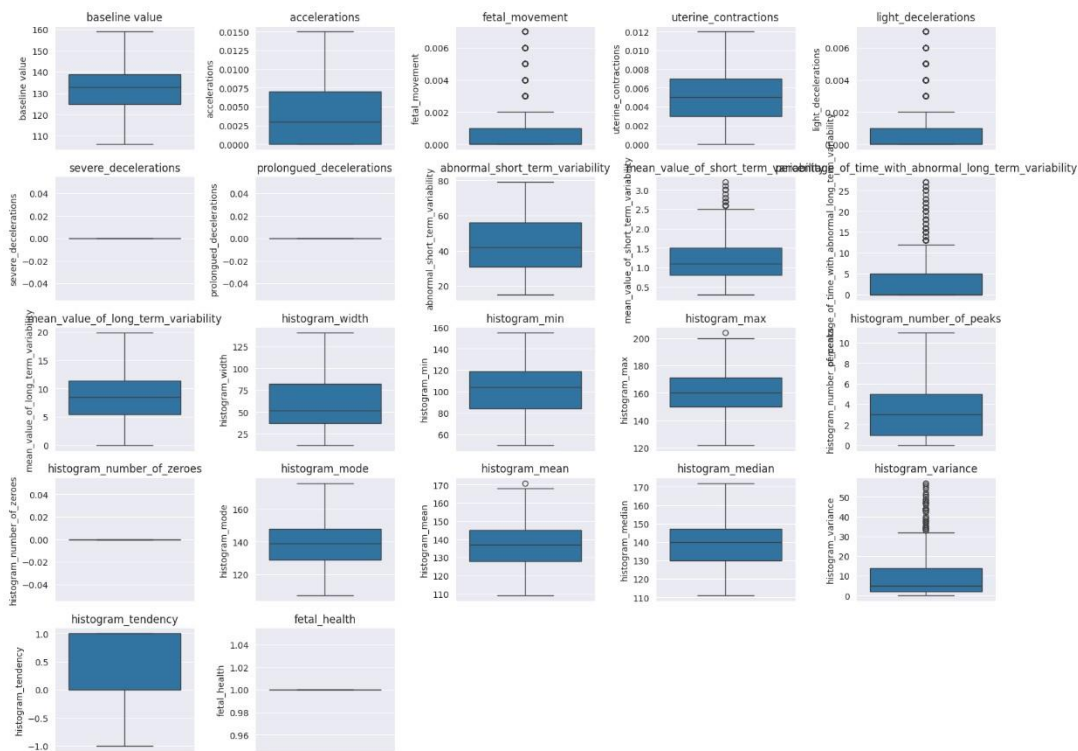
```
# PRINTING THE MAX OUTLIERS
plt.figure(figsize=(16, 12))
for i, column in enumerate(df.columns, 1):
    plt.subplot(5, 5, i)
    sns.boxplot(df[column])
    plt.title(column)
plt.tight_layout()
plt.show()
```



```
[63] # Calculate Q1 and Q3 for all columns
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
lower_limit = Q1 - 1.5 * IQR
upper_limit = Q3 + 1.5 * IQR
df_filtered = df[~((df < lower_limit) | (df > upper_limit)).any(axis=1)]
```

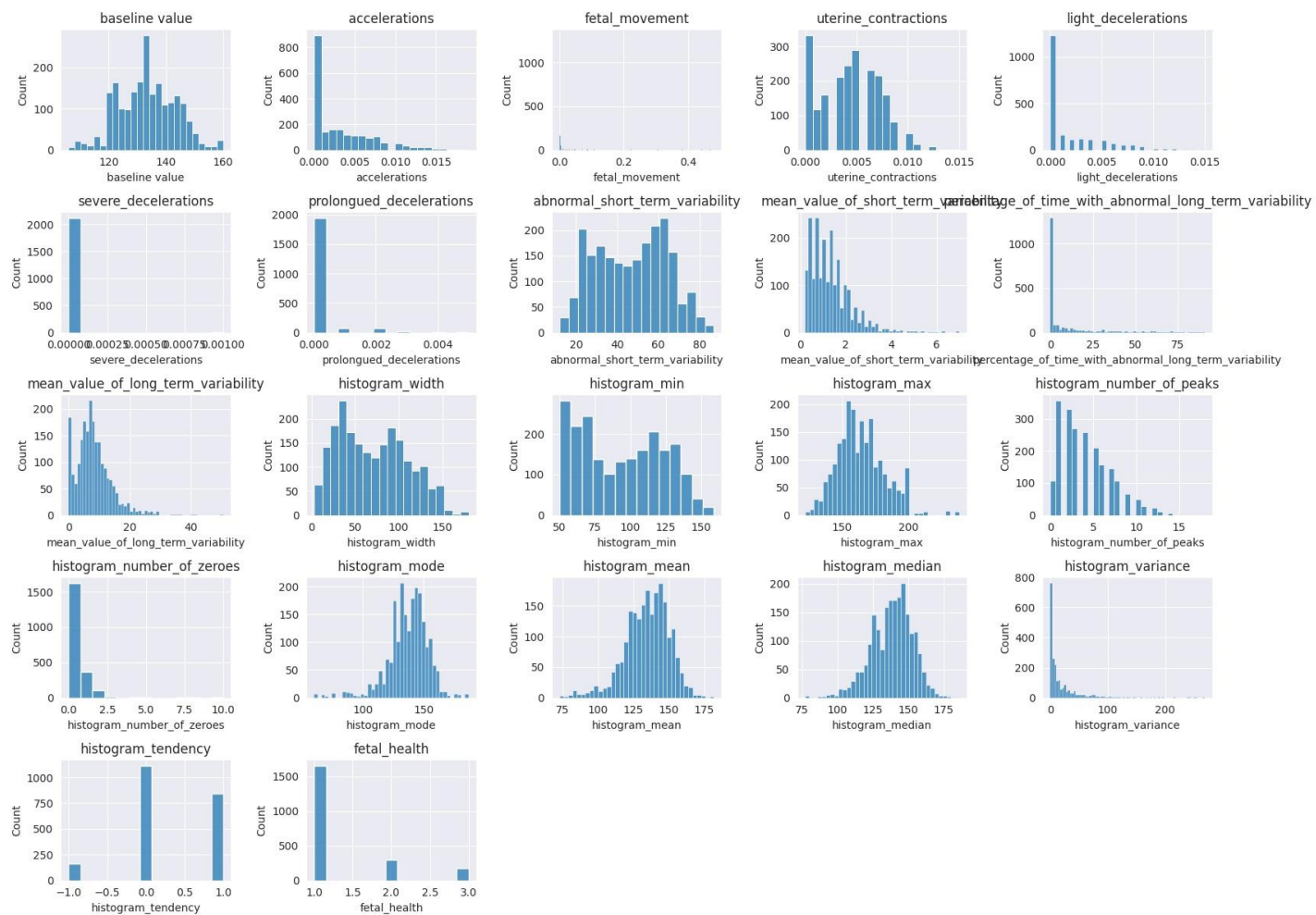
Printing boxplots for the filtered DataFrame

```
plt.figure(figsize=(16, 12))
for i, column in enumerate(df_filtered.columns, 1):
    plt.subplot(5, 5, i)
    sns.boxplot(df_filtered[column])
    plt.title(column)
plt.tight_layout()
plt.show()
```



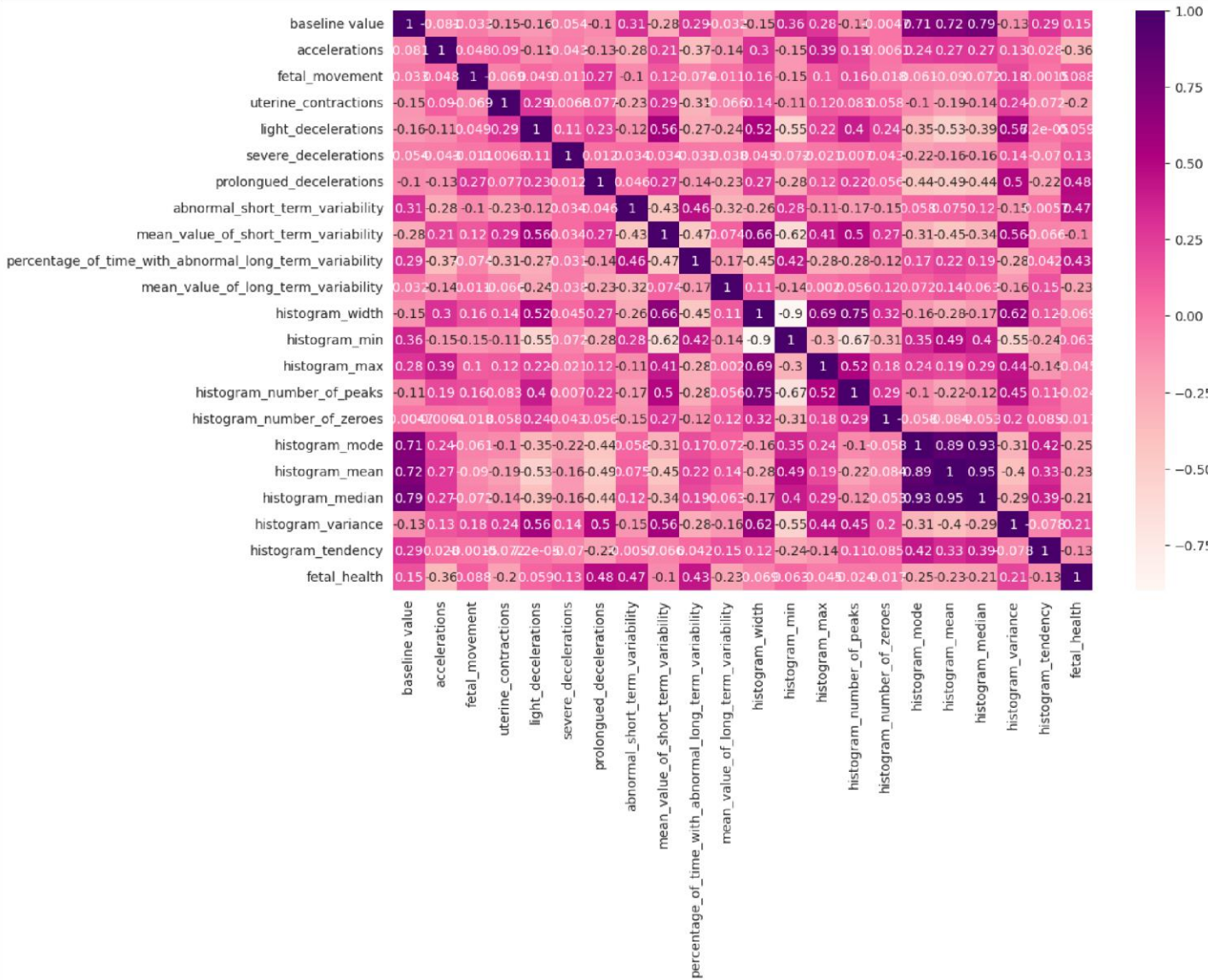
Visualizing Data

```
plt.figure(figsize=(16, 12))
for i, column in enumerate(df.columns, 1):
    plt.subplot(5, 5, i)
    sns.histplot(df[column], kde=False,)
    plt.title(column)
plt.tight_layout()
plt.show()
```



Heat Map

```
# generate heatmap to display correlations in data
corr = df.corr()
fig, ax = plt.subplots(figsize=(12, 8))
sns.heatmap(corr, annot=True, cmap="RdPu");
```



splitting data into train and test

```
[67] from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
[68] from sklearn.model_selection import train_test_split
x = df.drop('fetal_health', axis=1) # Replace 'target_variable' with your actual column name
y = df['fetal_health']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
x_train.shape, x_test.shape
```

```
((1700, 21), (426, 21))
```

```
pip install imblearn
```

```
Requirement already satisfied: imblearn in /usr/local/lib/python3.10/dist-packages (0.0)
Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.10/dist-packages (from imblearn) (0.10.1)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn->imblearn) (1.25.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn->imblearn) (1.11.4)
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn->imblearn) (1.2.2)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn->imblearn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn->imblearn) (3.5.0)
```

```
[70] y_train.shape, y_test.shape
```

```
((1700,), (426,))
```

```
from imblearn.over_sampling import SMOTE
smote=SMOTE()
print(y_train.unique())
# Check the number of unique classes in y_train
print("Number of unique classes in y_train:", len(y_train.unique()))
if len(y_train.unique()) > 1:
    x_train_smote, y_train_smote = smote.fit_resample(x_train.astype('float'), y_train)
else:
    print("SMOTE cannot be applied: y_train contains only one class.")
```

```
[1 2 3]
Number of unique classes in y_train: 3
```

```
[72] from imblearn.over_sampling import SMOTE
smote=SMOTE()
print(y_train.unique())
x_train_smote, y_train_smote = smote.fit_resample(x_train.astype('float'), y_train)
```

```
[1 2 3]
```

```
[73] from collections import Counter
print("before smote :", Counter(y_train))
print("after smote :", Counter(y_train_smote))
```

```
before smote : Counter({1: 1329, 2: 237, 3: 134})
after smote : Counter({1: 1329, 2: 1329, 3: 1329})
```

Model Building

Random Forest

```
[74] from sklearn.ensemble import RandomForestClassifier
```

```
rf=RandomForestClassifier()
rf.fit(x_train_smote, y_train_smote)
```

```
RandomForestClassifier()
RandomForestClassifier()
```

Accuracy Score of Random Forest

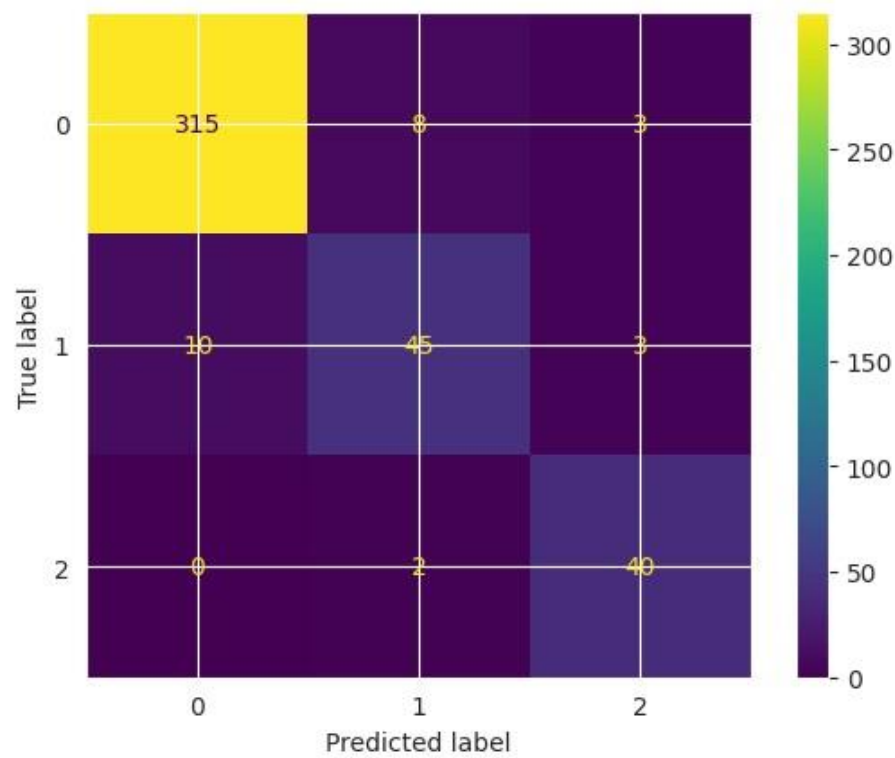
```
pred_rf.predict(x_test)
print("accuracy score")
print(accuracy_score(y_test,pred))
print("\n")
print("confusion matrix")
print(classification_report(y_test,pred))
print(confusion_matrix(y_test,pred))
```

accuracy score
0.9389671361502347

confusion matrix	precision	recall	f1-score	support
1	0.97	0.97	0.97	326
2	0.82	0.78	0.80	58
3	0.87	0.95	0.91	42
accuracy			0.94	426
macro avg	0.89	0.90	0.89	426
weighted avg	0.94	0.94	0.94	426

```
[[315  8  3]
 [ 10 45  3]
 [  0  2 40]]
```

```
[77] from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm=confusion_matrix(y_test,pred)
cm_display=ConfusionMatrixDisplay(cm).plot()
plt.show()
```



Decision Tree

```
[78] from sklearn.tree import DecisionTreeClassifier
```

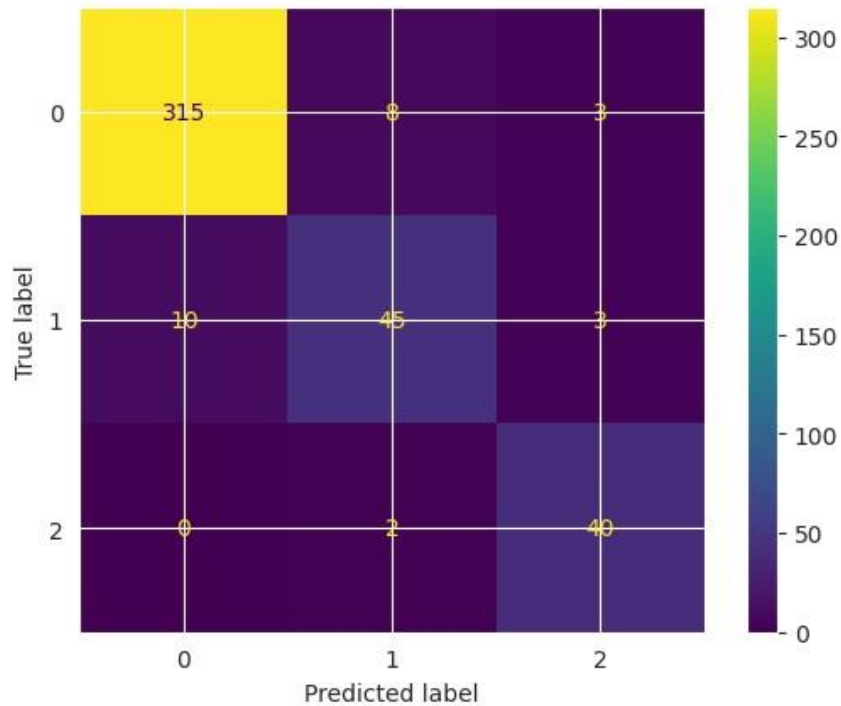
```
dt=DecisionTreeClassifier()  
dt.fit(x_train_smote,y_train_smote)  
pred=dt.predict(x_test)  
print("accuracy score")  
print(accuracy_score(y_test,pred))  
print("\n")  
print("confusion matrix")  
print(classification_report(y_test,pred))
```

accuracy score
0.9272300469483568

confusion matrix

	precision	recall	f1-score	support
1	0.97	0.96	0.96	326
2	0.73	0.81	0.77	58
3	0.92	0.86	0.89	42
accuracy			0.93	426
macro avg	0.87	0.87	0.87	426

```
cm_display=ConfusionMatrixDisplay(cm).plot()  
plt.show()
```



Logistic Regression

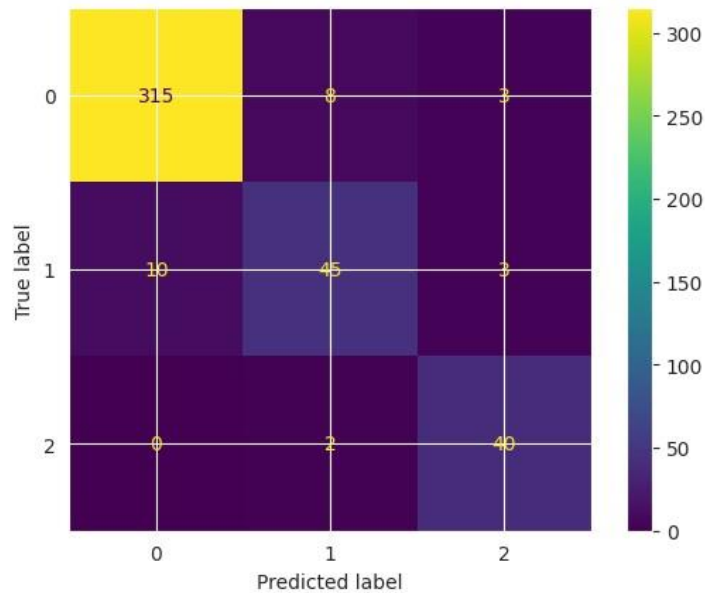
```
[81] from sklearn.linear_model import LogisticRegression
```

```
lr=LogisticRegression()
lr.fit(x_train_smote,y_train_smote)
pred=lr.predict(x_test)
print("accuracy score")
print(accuracy_score(y_test,pred))
print("\n")
print("confusion matrix")
```

```
accuracy score
0.7511737089201878
```

```
confusion matrix
```

```
[83] cm_display=ConfusionMatrixDisplay(cm).plot()
plt.show()
```



k-Nearest Neighbors

```
[84] from sklearn.neighbors import KNeighborsClassifier
```

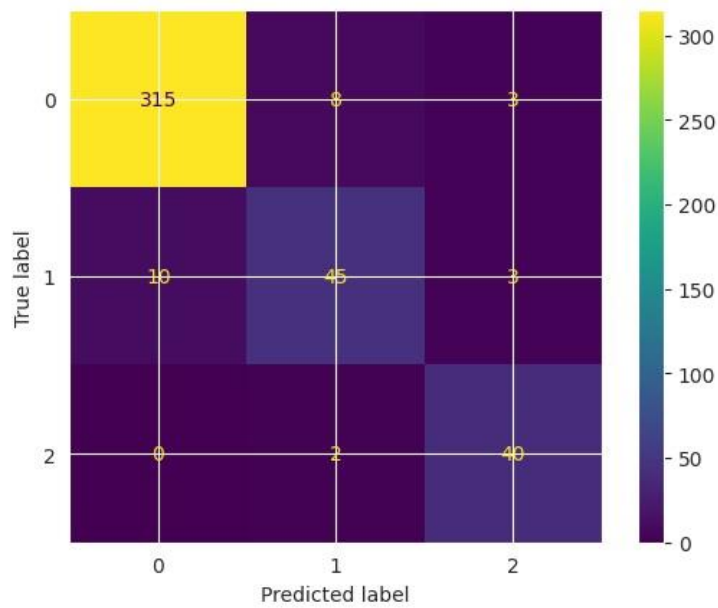
```
knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(x_train_smote,y_train_smote)
pred=knn.predict(x_test)
print("accuracy score")
print(accuracy_score(y_test,pred))
print("\n")
print("confusion matrix")
print(classification_report(y_test,pred))
```

```
accuracy score
0.8427230046948356
```

```
confusion matrix
```

	precision	recall	f1-score	support
1	0.96	0.86	0.90	326
2	0.49	0.79	0.61	58
3	0.85	0.81	0.83	42
accuracy			0.84	426
macro avg	0.76	0.82	0.78	426
weighted avg	0.80	0.84	0.86	426

```
cm_display=ConfusionMatrixDisplay(cm).plot()
plt.show()
```



```

Testing The Model
[87] test_data = [[120.0,0.000,0.000,0.000,0.000,0.0,0.0,73.0,0.5,0.43,0.62,0.126,0.2,0.0,120.0,137.0,121.0,73.0,1.0,2.0]] # fill in the missing values
      predictions = rf.predict(test_data)
      print(predictions)
[1]

[88] test_data=[[134.0 ,0.003 ,0.000 ,0.000, 0.003, 0.0 ,0.0 ,16.0 ,2.4, 0.0 ,0.53,0 ,170.0 ,11.0,0.0,137.0,134.0,137.0,13.0,1.0,1.0]]
      pred=rf.predict(test_data)
      print(pred)
[1]

Deployment
[89] import pickle
      pickle.dump(knn,open('fetal_health3.pkl','wb'))
  
```

INDEX.HTML

```

<!DOCTYPE html> <html
lang="en">
  
```

```

<head>
  
```

```

    <meta charset="utf-8">
  
```

```

    <meta content="width=device-width, initial-scale=1.0" name="viewport">
  
```

```

    <title>Index - Fetal_AI Bootstrap Template</title>
  
```

```

<meta content="" name="description">
<meta content="" name="keywords">

<!-- Favicons -->
<link href="../static/im/favicon.png" rel="icon">
<link href="../static/img/apple-touch-icon.png" rel="apple-touch-icon">

<!-- Fonts -->
<link href="https://fonts.googleapis.com" rel="preconnect">
<link href="https://fonts.gstatic.com" rel="preconnect" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;0,400;0,500;0,600;0,700;0,800;1,300;1,400;1,500;1,600;1,700;1,800&family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">

<!-- Vendor CSS Files -->
<link href="../static/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link href="../static/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
<link href="../static/vendor/aos/aos.css" rel="stylesheet">
<link href="../static/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
<link href="../static/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

<!-- Main CSS File -->
<link href="../static/css/main.css" rel="stylesheet">

<body class="index-page">

<header id="header" class="header d-flex align-items-center fixed-top">
<div class="container-fluid position-relative d-flex align-items-center justify-content-between">
<a href="index.html" class="logo d-flex align-items-center me-auto me-xl-0">
<!-- Uncomment the line below if you also wish to use an image logo -->
<!--  -->
<h1 class="sitename">Fetal_AI</h1><span>.</span>
</a>

<nav id="navmenu" class="navmenu">
<ul>
<li><a href="index.html#hero" class="active">Home</a></li>
<li><a href="index.html#about">About</a></li>
</li>

```

```

        <li><a href="index.html#contact">Contact</a></li>
    </ul>
    <i class="mobile-nav-toggle d-xl-none bi bi-list"></i>
</nav>

<a class="btn-getstarted" href="/predict">PREDICT NOW</a>

</div>
</header>

<main class="main">

    <!-- Hero Section -->
    <section id="hero" class="hero section dark-background">

        
        <div class="container">
            <div class="row">
                <div class="col-lg-10">
                    <h2 data-aos="fade-up" data-aos-delay="500">PREDICT AND MONITOR
FETAL HEALTH</h2>
                    <p data-aos="fade-up" data-aos-delay="200"> </p>

                </div>
            </div>
        </div>
    </div>

</section><!-- /Hero Section -->
    <!-- Clients Section -->
    <section id="clients" class="clients section">

</section><!-- /Clients Section -->

    <!-- About Section -->
    <section id="about" class="about section light-background">

        <div class="container" data-aos="fade-up" data-aos-delay="100">
            <div class="row align-items-xl-center gy-5">

                <div class="col-xl-5 content">
                    <h3>About us</h3>

```

<h2>What Is Fetal AI</h2>

<p>Reduction of child mortality is reflected in several of the United Nations' Sustainable Development Goals and is a key indicator of human progress. The UN expects that by 2030, countries end preventable deaths of newborns and children under 5 years of age, with all countries aiming to reduce under 5 mortality to at least as low as 25 per 1,000 live births. Parallel to 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). The vast majority of these deaths (94%) occurred in low-resource settings, and most could have been prevented. In light of what was mentioned above, Cardiotocograms (CTGs) are a simple and cost accessible option to assess fetal health, allowing healthcare professionals to take action in order to prevent child and maternal mortality. The equipment itself works by sending ultrasound pulses and reading its response, thus shedding light on fetal heart rate (FHR), fetal movements, uterine contractions and more. In this project, we have some characteristics of Fetal Health as a dataset. The target variable of this dataset is Fetal Health. Since it is a multi class classification, the classes are represented by 'Normal', 'Pathological' and 'Suspect'.</p>

Read More<i class="bi biarrow-right"></i>

</div>

<div class="col-xl-7">

<div class="row gy-2 icon-boxes">

<div class="col-md-22" data-aos="fade-up" data-aos-delay="200">

<div class="icon-box">

<i class="bi bi-buildings"></i>

<h3>Prototype </h3>

</div>

</div> <!-- End Icon Box -->

</div>

</div>

</section><!-- /Stats Section -->

<!-- upto here completed -->

<!-- Contact Section -->

<section id="contact" class="contact section">

<!-- Section Title -->

<div class="container section-title" data-aos="fade-up">

<h2>Contact Us</h2>

</div><!-- End Section Title -->


```
<div class="container" data-aos="fade-up" data-aos-delay="100">
```

```
<div class="row gy-4">
```

```
<div class="col-lg-6">
```

```
<div class="row gy-4">
```

```
<div class="col-md-6">
```

```
<div class="info-item" data-aos="fade" data-aos-delay="200">
```

```
<i class="bi bi-geo-alt"></i>
```

```
<h3>Address</h3>
```

```
<p>Vaagdevi Engineering College </p>
```

```
<p>Bollikunta X Road , Warangal </p>
```

```
</div>
```

```
</div><!-- End Info Item -->
```

```
<div class="col-md-6">
```

```
<div class="info-item" data-aos="fade" data-aos-delay="300">
```

```
<i class="bi bi-telephone"></i>
```

```
<h3>Call Us</h3>
```

```
<p>91+ 9010767269</p>
```

```
<p>91+ 9372682679</p>
```

```
</div>
```

```
</div><!-- End Info Item -->
```

```
<div class="col-md-6">
```

```
<div class="info-item" data-aos="fade" data-aos-delay="400">
```

```
<i class="bi bi-envelope"></i>
```

```
<h3>Email Us</h3>
```

```
<p>jashwanthboddupally@gmail.com</p>
```

```
<p>vaagdeviengineeringcollege.com</p>
```

```
</div>
```

```
</div><!-- End Info Item -->
```

```
<div class="col-md-6">
```

```
<div class="info-item" data-aos="fade" data-aos-delay="500">
```

```
<i class="bi bi-clock"></i>
```

```
<h3>Open Hours</h3>
```

```
<p>24/7</p>
```

```
<p>Any Time Available</p>
```

```
</div>
```

```
</div><!-- End Info Item -->
```

```
</div>
```

</div>

<div class="col-lg-6">

<form action="forms/contact.php" method="post" class="php-email-form"
data-aos="fade-up" data-aos-delay="200">

<div class="row gy-4">

<div class="col-md-6">

<input type="text" name="name" class="form-control"
placeholder="Your Name" required="">

</div>

<div class="col-md-6 ">

<input type="email" class="form-control" name="email"
placeholder="Your Email" required="">

</div>

<div class="col-12">

<input type="text" class="form-control" name="subject"
placeholder="Subject" required="">

</div>

<div class="col-12">

<textarea class="form-control" name="message" rows="6"
placeholder="Message" required=""></textarea>

</div>

<div class="col-12 text-center">

<div class="loading">Loading</div>

<div class="error-message"></div>

<div class="sent-message">Your message has been sent. Thank
you!</div>

<button type="submit">Send Message</button>

</div>

</div>

</form>

</div><!-- End Contact Form -->

</div>

</div>

</section><!-- /Contact Section -->

</main>

<footer id="footer" class="footer position-relative light-background">

<div class="container footer-top">

<div class="row gy-4">

<div class="col-lg-5 col-md-12 footer-about">

Fetal_AI

<p>Fetal AI, also known as fetal artificial intelligence, refers to the application of artificial intelligence (AI) techniques and technologies in the field of prenatal care and fetal health monitoring. This emerging field holds significant promise for improving the monitoring, diagnosis, and management of fetal health conditions during pregnancy. Here are some key aspects and implications of fetal AI.</p>

<div class="social-links d-flex mt-4">

<i class="bi bi-twitter-x"></i>

<i class="bi bi-facebook"></i>

<i class="bi bi-instagram"></i>

<i class="bi bi-linkedin"></i>

</div>

</div>

<div class="col-lg-2 col-6 footer-links">

<h4>Useful Links</h4>

Home

About us

Privacy policy

</div>

<div class="col-lg-3 col-md-12 footer-contact text-center text-md-start">

<h4>Contact Us</h4>

<p>Vaagdevi Engineering College</p>

<p>Bollikunta , Warangal</p>

<p>INDIA </p>

<p class="mt-4">Phone: +91 9010767269</p>

<p>Email: jashwanthboddupally.com</p>

</div>

```

    </div>
</div>

<div class="container copyright text-center mt-4">
    <p>© <span>Copyright</span> <strong class="sitename">Fetal_AI</strong>
<span>All Rights Reserved</span></p>
    <div class="credits">
        <!-- All the links in the footer should remain intact. -->
        <!-- You can delete the links only if you've purchased the pro version. -->
        <!-- Licensing information: https://bootstrapmade.com/license/ -->
        <!-- Purchase the pro version with working PHP/AJAX contact form: [buy-url]
-->
        Designed by <a href="https://bootstrapmade.com/">BootstrapMade</a>
    </div>
</div>

</footer>

<!-- Scroll Top -->
<a href="#" id="scroll-top" class="scroll-top d-flex align-items-center
justifycontent-center"><i class="bi bi-arrow-up-short"></i></a>

<!-- Preloader --> <div
id="preloader"></div>

<!-- Vendor JS Files -->
<script src="../static/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="../static/vendor/php-email-form/validate.js"></script>
<script src="../static/vendor/aos/aos.js"></script>
<script src="../static/vendor/glightbox/js/glightbox.min.js"></script>
<script src="../static/vendor/purecounter/purecounter_vanilla.js"></script>
<script src="../static/vendor/imagesloaded/imagesloaded.pkgd.min.js"></script>
<script src="../static/vendor/isotope-layout/isotope.pkgd.min.js"></script>
<script src="../static/vendor/swiper/swiper-bundle.min.js"></script>

<!-- Main JS File -->
<script src="../static/js/main.js"></script>

</body>

</html>

```

PREDICT.HTML

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Fetal Checkup</title>
```

```
    <style>
```

```
        body {
            font-family: Arial, sans-serif;
display: flex;            justify-content:
center;            align-items: center;
height: 100vh;
            margin: 0;
            background: url('https://example.com/background-image.jpg') no-repeat
center center fixed;
            background-size: cover;
        }
        .container {
            background: rgba(178, 233, 202, 0.8);
border: 2px solid #000000;            border-
radius: 20px;            padding: 30px;
            box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);
width: 100%;            max-width: 1200px;
            box-sizing: border-box;
        }
        h1 {
            text-align: center;
color: #fa3f3f;
        }
        .conditions {
            margin-
bottom: 20px;            padding:
10px;            border: 1px solid
#00BFFF;            border-radius:
5px;
            background-color: #f0f8ff;
        }
        .conditions h2 {
text-align: center;
margin: 0;            color:
#333;

```

```

    }
    .inputs {
display: grid;
    grid-template-columns: repeat(3, 1fr);
    gap: 30px;
    }
    .input-group {
display: flex;        flex-
direction: column;
    }
    .input-group label {        margin-
bottom: 5px;        font-weight: bold;
    color: #333;
    }
    .input-group input {
padding: 10px;
border: 1px solid #ccc;
border-radius: 5px;
    box-sizing: border-box;
    }
    .submit-btn {
display: flex;        justify-
content: center;
    margin-top: 20px;
    }
    .submit-btn button {
padding: 10px 20px;        border:
none;        border-radius: 5px;
background-color: #00BFFF;
color: white;        font-size: 16px;
cursor: pointer;
    }
    .submit-btn button:hover {
    background-color: #009acd;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Fetal Health CheckUp</h1>
        <div class="conditions">
            <h2>FETAL HEALTH</h2>
            </div>
        <form action="/output" method="POST">

```

```
<div class="inputs">
  <div class="input-group">
    <label for="text1">prolongued_decelerations </label>
    <input type="number" id="text1" placeholder="Enter number">
  </div>
  <div class="input-group">
    <label for="text2">histrogram_variance</label>
    <input type="number" id="text2" placeholder="Enter number">
  </div>
  <div class="input-group">
    <label for="text3">histrogram_mode</label>
    <input type="number" id="text3" placeholder="Enter number">
  </div>
  <div class="input-group">
    <label for="text4">abnormal_short_term_variability</label>
    <input type="number" id="text4" placeholder="Enter number">
  </div>
  <div class="input-group">
    <label for="text5">histrogram_median</label>
    <input type="number" id="text5" placeholder="Enter number">
  </div>
  <div class="input-group">
    <label for="text6">accelerations</label>
    <input type="number" id="text6" placeholder="Enter number">
  </div>
  <div class="input-group">
    <label
for="text7">percentage_of_time_with_abnormal_long_term_variability</label>
    <input type="number" id="text7" placeholder="Enter number">
  </div>
  <div class="input-group">
    <label for="text8">mean_value_of_long_term_variability</label>
    <input type="number" id="text8" placeholder="Enter number">
  </div>
</div>
<div class="submit-btn">
  <button type="submit">SUBMIT</button>
</div>
</form>
</div>
</body> </html>
```

APP.PY

```
from flask import Flask,request,render_template
import numpy as np
import pandas as pd
import pickle

model=pickle.load(open(r'C:\Users\jashwanth\Documents\Fetal_AI\Fetal_AI\fetal_
_health1.pkl','rb'))
app=Flask(__name__)
@app.route("/") def home():    return
render_template("index.html")

# ROUTING
@app.route('/predict',methods=["GET","POST"])
def predict():    render_template('predict.html')

@app.route("/output",methods=["GET","POST"])
def home():
    prolonged_decelerations = float(request.form['prolongued_decelerations'])
    abnormal_short_term_variability=
float(request.form['abnormal_short_term_variability'])
    percentage_of_time_with_abnormal_long_term_variability =
float(request.form['percentage_of_time_with_abnormal_long_term_variability'])
    histogram_variance = float(request.form['histogram_variance'])
    histogram_median =float(request.form['histogram_median'])
    mean_value_of_long_term_variability =
float(request.form['mean_value_of_long_term_variability'])
    histogram_mode =
float(request.form['histogram_mode'])    accelerations =
float(request.form['accelerations'])    x =
[[prolongued_decelerations,abnormal_short_term_variability,percentage_of_time_
with_abnormal_long_term_variability,]]

    output = model.predict(x)
    out=['NORMAL','PATHOLOGICAL','SUSPECT']
    if int(output[0])==0:    output='NORMAL'    elif
int(output[0])==1:
        output='PATHOLOGICAL'
    else:
        output='SUSPECT'

    return render_template('output.html',output=output)
```



```
#MAIN FUNCTION : if  
__name__ == "__main__":  
app.run(debug=True)
```

10.2 GitHub and project Demo link:

Github link:[Click Here](#)

Project Demo link:[Click Here](#)