



Kabul University

Computer Science Faculty

Information Systems Department

Predicting Emergency Situation of Mother and Baby in Skill Delivery by Machine learning

A monograph submitted in partial fulfillment of the requirements

for the award of the degree of

Bachelor of Information Systems in Computer Science

Submitted by:

Qadir Ali “Adalat”

Supervised by:

Asst.Prof. Hedayat “Lodin”

November, 2023

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Certificate of Approval

I certify that I have read the “Predicting Emergency Situation of Mother and Baby in Skill Delivery by Machine Learning” report submitted by Qadir Ali “Adalat” as partial fulfillment for the award of the degree of Bachelor of Information Systems Faculty of Computer Science at Kabul University. I have evaluated the report and found it up to the requirements in its scope and quality for the award of the degree.

Supervisor Name: Asst. Prof. Hedayat “Lodin”

Signature: _____

Date: _____

Name: Associate. Prof: Mohammad Shuaib “Zarinkhil”

(Dean of Information System Department)

Signature: _____

Date: _____

Name: Associate. Prof. Amir Krar “Shahidzay”

(Dean of Computer Science Faculty)

Signature: _____

Date: _____

Author's Declaration

Hereby I, Qadir Ali “Adalat”, declare that this Bachelor thesis is my personal achievement and the result of an original investigation. I further state that this thesis has not been previously submitted for any other academic degree.

Date: November -2023

B.CS. Student Name and Sign: _____

Supervisor Name and Sign: _____

Abstract

This study addresses the urgent need for improved healthcare outcomes in Afghanistan, specifically focusing on the high maternal and infant mortality rates and limited access to skilled birth attendants and healthcare facilities. The objective of this research is to develop a predictive model using machine learning techniques to identify emergency situations during childbirth and provide early warnings to mitigate risks.

Six different machine learning algorithms, namely K-Nearest Neighbors, Support Vector Machines, Random Forest, Neural Networks, Logistic Regression, and Naive Bayes, were evaluated for their performance using metrics such as accuracy, precision, recall, and F1-score. After a comprehensive evaluation, the Random Forest algorithm emerged as the optimal choice due to its high accuracy (93%) and reliable results.

The proposed solution involves deploying the predictive model in both urban and rural settings. In urban areas, the model can be utilized in hospitals before delivery to predict whether a mother will have a normal delivery or require emergency interventions. This enables healthcare providers to take appropriate actions and provide timely care. Additionally, pregnant mothers at or beyond 30 weeks' gestation can input their information for stage-specific predictions. In rural villages with limited access to hospitals and resources, the model becomes invaluable. By using the model, mothers can determine if an emergency situation is likely, thereby avoiding unnecessary and risky journeys to distant hospitals. Instead, they can safely deliver with the assistance of midwives in their villages.

By leveraging computational tools and advanced analytics, this study aims to contribute to the reduction of maternal and infant mortality rates in Afghanistan. The predictive model offers an innovative approach to identify high-risk cases, enabling timely interventions and more efficient allocation of healthcare resources. The comprehensive documentation provides insights into the project's architecture, data preprocessing techniques, model training processes, and outcomes achieved using the Random Forest algorithm. The results highlight the potential of machine learning in improving healthcare outcomes and addressing the unique challenges faced by Afghanistan in maternal and infant healthcare.

Key words: maternal and infant mortality, predictive model, machine learning, healthcare outcomes, Afghanistan

Acknowledgment

I am profoundly thankful for the chance to undertake this project as part of our academic path in Predicting Emergency Situation of Mother and Baby in Skill Delivery by Machine Learning at Kabul University's Information Systems department in Afghanistan. My sincere gratitude goes to my project supervisor, Assistant Professor Hedayat “Lodin”, for his consistent guidance, support, and commitment, which significantly influenced my project's trajectory. I also extend my appreciation to my classmates and families for their essential editing help and unwavering moral support, serving as a perpetual source of motivation during the project's development.

Acronyms

PC	Personal Computer
WHO	World Health Organization
MoPH	Ministry of Public Health
KNN	K Nearest Neighbor
SPV	Support Vector Machine
NN	Neural Network
NB	Naive Bayes
RF	Random Forest
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
ROC	Receiver Operating Characteristic Curve

Table of Contents

List of Figures	X
Chapter 1: Introduction	1
1.1 Introduction	1
1.2 Problem statement	2
1.3 Importance of the problem	2
1.4 Aims & objectives	3
1.5 Methodology.....	3
1.5.1 CRSP Methodology	3
1.5.2 Project Scope	4
1.5.3 Expected outcome.....	5
Chapter 2: Related Work (Literature Review)	7
Chapter 3: Research Methodology.....	10
3.1 Data Gathering.....	10
3.2 Preprocessing Steps	11
3.2.1 Handling Missing Values	11
3.2.2 Data Scaling - Standardization	11
3.3 Algorithm Selection and Evaluation.....	12
3.3.1 K-Nearest Neighbors (KNN).....	12
3.3.2 Support Vector Machines (SVM).....	14
3.3.3 Random Forest (RF)	15
3.3.4 Neural Networks (NN)	17
3.3.5 Logistic Regression	19
3.3.6 Naive Bayes (NB).....	20
3.4 Evaluation metrics for each algorithm.....	23
3.4.1 Accuracy:.....	23
3.4.2 Precision:	24
3.4.3 Recall:.....	26
3.4.4 F1-score:	27
3.4 Selected Model	29
3.5 Making Predictions on New Data.....	30
Chapter 4: Implementation Details	32

4.1 Dataset Loading:.....	32
4.2 Data Preprocessing:	32
4.2.1 Handling Missing Values:	33
4.2.2 Splitting the Dataset:	33
4.3 Feature Extraction:	33
4.4 Data Scaling.....	33
4.5 Model Training:.....	34
4.6 Prediction Function:	34
4.7 Interface Development	35
4.8 To use the Emergency Situation Prediction system	36
4.8.1 Submit the form to initiate the prediction process:.....	36
4.9 The project code consists of the following components.....	37
4.9.1 app.py:	37
4.9.2 Main_page.html:.....	37
4.9.3 result.html:.....	37
4.10 Getting Started.....	37
Chapter 5: Results	40
5.1 Limitation	41
5.1.1 Limited availability of hospital data:	41
5.1.2 Challenges in data collection from patient documents	41
5.1.3 Difficulty in data collection from women's hospitals	41
5.1.4 Ethical considerations.....	41
5.2 Future Work.....	42
5.2.1. Development of a mobile application:.....	42
5.2.2 Offline functionality:	42
5.2.3 Continuous data collection and model refinement:	42
5.2.4 Evaluation in real-world settings:	42
Conclusion	43
References	45

List of Figures

Figure 1. CRSP Methodology in my project.....	4
Figure 2. General Dataset	10
Figure 3. Separated Dataset.....	10
Figure 4. Implementation of the predictive model	12
Figure 5. Model Evaluation.....	29
Figure 6. Dataset loading.....	31
Figure 7. Data preprocessing and handling missing values.....	31
Figure 8. Feature Extraction.....	32
Figure 9. Data Scaling and standardization.....	32
Figure 10. Training RF model.....	33
Figure 11. Predict function.....	33
Figure 12. Interface.....	35

Chapter 1: Introduction

1.1 Introduction

Afghanistan faces a critical challenge with its high maternal and infant mortality rates, which are further exacerbated by limited access to skilled birth attendants and healthcare facilities. The World Health Organization (WHO) and UNICEF have reported alarming statistics highlighting the urgent need for improved healthcare outcomes in the country. In response to this pressing issue, this research aims to develop a predictive model using machine learning techniques to identify emergency situations during childbirth and provide early warnings to mitigate risks (WHO, 2021; UNICEF, 2021). [1][2]

The primary objective of this study is to leverage the power of machine learning algorithms to predict emergency situations and provide healthcare professionals with timely information for appropriate interventions. By implementing a predictive model, healthcare providers can proactively identify high-risk cases and allocate resources effectively. This can lead to significant improvements in maternal and infant healthcare outcomes.

To achieve this objective, six different machine learning algorithms, namely K-Nearest Neighbors, Support Vector Machines, Random Forest, Neural Networks, Logistic Regression, and Naive Bayes, were evaluated for their performance. Various metrics such as accuracy, precision, recall, and F1-score were used to assess the effectiveness of these algorithms. Based on the comprehensive evaluation, the Random Forest algorithm was identified as the optimal choice, exhibiting high accuracy and reliable results.

The proposed solution involves deploying the predictive model in both urban and rural settings. In urban areas, the model can be integrated into existing healthcare systems to predict whether a mother will require emergency interventions during delivery. This allows healthcare providers to take appropriate actions and provide timely care. Additionally, pregnant mothers at or beyond 30 weeks' gestation can utilize the model to receive stage-specific predictions based on their individual information.

In rural villages where access to hospitals and resources is limited, the predictive model becomes particularly valuable. Mothers can input their data into the model to assess the likelihood of an emergency situation during childbirth. This empowers them to make informed decisions and avoid unnecessary and

risky journeys to distant hospitals. Instead, they can safely deliver with the assistance of midwives in their villages, ensuring better outcomes for both mothers and infants.

By combining computational tools and advanced analytics, this study aims to contribute to the reduction of maternal and infant mortality rates in Afghanistan. The predictive model offers an innovative approach to identify high-risk cases, enabling timely interventions and more efficient allocation of healthcare resources. Through comprehensive documentation, this research provides insights into the project's architecture, data preprocessing techniques, model training processes, and outcomes achieved using the Random Forest algorithm.

In the subsequent sections of this monograph, we will delve into the details of the research methodology, implementation, results, and discussions. The conclusion will summarize the key findings, limitations, and future directions for further research. By leveraging the potential of machine learning, this study aspires to make substantial progress in improving healthcare outcomes and addressing the unique challenges faced by Afghanistan in maternal and infant healthcare.

1.2 Problem statement

Afghanistan has one of the highest maternal mortality rates in the world, with a significant number of deaths occurring during childbirth. One of the main reasons for this is the shortage of skilled childbirth delivery services in the country. Despite efforts by the government and other organizations to improve maternal health outcomes, the lack of access to skilled childbirth delivery services remains a significant challenge.

One of the key challenges in skilled childbirth delivery in Afghanistan is the shortage of trained healthcare professionals, particularly midwives. Many healthcare facilities lack the necessary staff and resources to provide quality maternal healthcare services, particularly in rural areas. Additionally, there is a significant gender gap in the healthcare workforce, with few female healthcare professionals available to provide care to women during childbirth. Another challenge is the lack of access to quality maternal healthcare services, particularly in remote and conflict-affected areas. Many women in these areas do not have access to healthcare facilities or trained healthcare professionals, which increases the risk of maternal mortality and morbidity.

1.3 Importance of the problem

The shortage of skilled childbirth delivery services in Afghanistan is a major challenge that contributes to the high maternal mortality rate in the country, particularly in rural and conflict-affected areas. The shortage of trained healthcare professionals, especially midwives, and the lack of access to quality

maternal healthcare services puts the lives of women at risk and has far-reaching consequences for families and communities. Addressing this problem is crucial for improving maternal health outcomes development in Afghanistan.

1.4 Aims & objectives

Due to the problem I mentioned, I have decided to create a predictive model using machine learning to provide early warnings of mother and baby mortality. This model can be used in both cities and villages. In cities, we can use it in hospitals before delivery. When the mother comes to the hospital for delivery, we can use the model to predict whether the mother will have a normal delivery or if an emergency situation is likely. This model is also useful for mothers who are 30 weeks or more pregnant. They can provide all necessary information to the model, including gestational age, and the model will predict that specific gestational age.

The model is particularly beneficial in villages where there are no good hospitals or proper equipment, and the distance to the hospital is too far. It is difficult for mothers to come to the hospital every week for checkups, and some villages are more than 5 hours away from the nearest hospital. If they move the mother to the hospital, it may be too late, and it could potentially harm the mother or baby. By using this model, if the model predicts "no emergency," they do not need to come to the hospital. Instead, the mother can deliver with the assistance of a midwife.

1.5 Methodology

1.5.1 CRSP Methodology

CRISP-DM stands for cross-industry process for data mining. The CRISP-DM methodology provides a structured approach to planning a data mining project. It is a robust and well-proven methodology. We do not claim any ownership over it. We did not invent it. We are however evangelists of its powerful practicality, its flexibility and its usefulness when using analytics to solve thorny business issues. It is the golden thread that runs through almost every client engagement. The CRISP-DM model is shown on the right. [9]

1. Business understanding
2. Data understanding
3. Data preparation

4. Modeling
5. Evaluation
6. Deployment

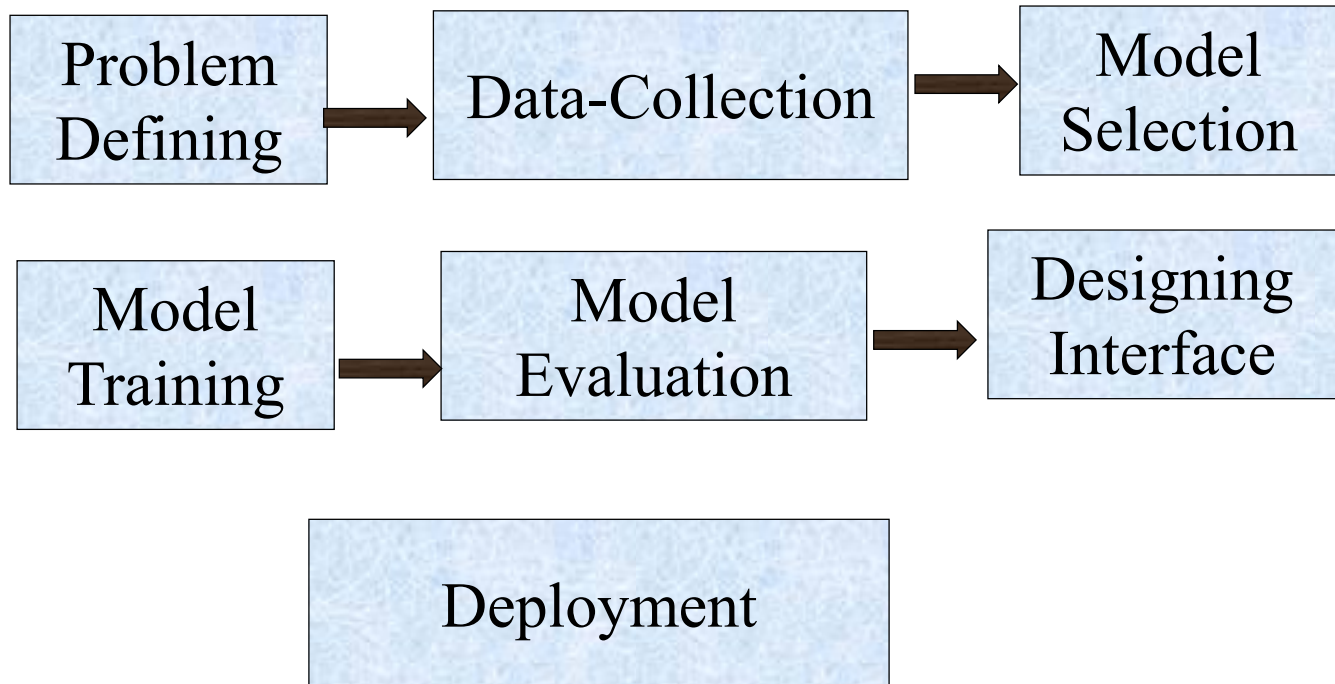


Figure [1] **CRSP Methodology in my project**

1.5.2 Project Scope

Objective

The objective of this project is to develop a predictive model using machine learning techniques to identify emergency situations during childbirth, with a focus on improving healthcare outcomes for mothers and infants in Afghanistan.

Scope

Addressing high maternal and infant mortality rates: The project aims to tackle the critical challenge of high maternal and infant mortality rates in Afghanistan. It focuses on developing a predictive model that can provide early warnings of emergency situations during childbirth.

Machine learning model development and evaluation: The project includes the development and evaluation of a predictive model using six different machine learning algorithms: K-Nearest Neighbors, Support Vector Machines, Random Forest, Neural Networks, Logistic Regression, and Naive Bayes. The

performance of these algorithms will be assessed using metrics such as accuracy, precision, recall, and F1-score.

Model in urban and rural settings: The scope extends to both urban and rural areas of Afghanistan. In urban areas, the model can be implemented in hospitals to predict the likelihood of normal deliveries or emergencies when mothers present for delivery. It can also provide stage-specific predictions for pregnant mothers at or beyond 30 weeks' gestation. In rural areas with limited access to healthcare resources, the model becomes particularly valuable, allowing mothers to assess the likelihood of emergency situations and make informed decisions about seeking medical assistance.

Documentation and outcomes: The documentation includes comprehensive documentation of the project, covering aspects such as the project's architecture, data preprocessing techniques, model training processes, and outcomes achieved using the Random Forest algorithm. The documentation aims to provide insights into the methodology and highlight the potential of machine learning in improving maternal and infant healthcare in Afghanistan.

It's important to note that this project scope focuses on the development and evaluation of the predictive model and its application within the given context. It does not address the underlying socio-economic factors or structural challenges contributing to high maternal and infant mortality rates in Afghanistan.

1.5.3 Expected outcome

The expected outcome of this project is to develop a predictive model for identifying emergency situations during childbirth that can significantly contribute to improving healthcare outcomes for mothers and infants in Afghanistan. Specifically, the following outcomes are anticipated:

1. Identification of the optimal algorithm: Through the evaluation of six different machine learning algorithms, including K-Nearest Neighbors, Support Vector Machines, Random Forest, Neural Networks, Logistic Regression, and Naive Bayes, the research aims to identify the algorithm that performs best in predicting emergency situations during childbirth. The expected outcome is to determine the most accurate and reliable algorithm for the given context.

2. Deployment of the predictive model: The project intends to deploy the developed predictive model in both urban and rural settings. In urban areas, the model can be utilized in hospitals to predict the likelihood of normal deliveries or emergencies, enabling healthcare providers to take appropriate actions and provide timely care. In rural villages with limited access to healthcare resources, the model becomes invaluable, allowing mothers to assess the likelihood of emergency situations and safely deliver with the assistance of midwives in their villages. The expected outcome is the successful implementation and utilization of the predictive model in real-world healthcare settings.

3. Improved healthcare outcomes: By providing early warnings and timely interventions, the predictive model aims to contribute to the reduction of maternal and infant mortality rates in Afghanistan. The expected outcome is to demonstrate improved healthcare outcomes, including a decrease in emergency situations during childbirth, better allocation of healthcare resources, and ultimately, a positive impact on the overall maternal and infant health in the country.

4. Documentation and knowledge sharing: The project aims to document the project's architecture, data preprocessing techniques, model training processes, and outcomes achieved using the Random Forest algorithm. The expected outcome is to provide comprehensive documentation that can serve as a valuable resource for future studies and initiatives in the field of maternal and infant healthcare, showcasing the potential of machine learning in addressing the unique challenges faced by Afghanistan. Overall, the expected outcome of this research is to contribute to the advancement of healthcare practices in Afghanistan by developing an effective predictive model for identifying emergency situations during childbirth and improving the overall health outcomes for mothers and infants in the country.

Chapter 2: Related Work (Literature Review)

Skilled assistance during child delivery is essential to reduce maternal mortality. Several studies have been conducted to determine the importance of skilled child delivery services in countries like Afghanistan and Ethiopia. In Afghanistan, which has a high rate of maternal mortality and low coverage of skilled birth attendants, a study aimed to implement a predictive model using data mining classification algorithms. The objective was to efficiently use scarce skilled child delivery services in the country. The study utilized a binary classification approach and five popular machine learning techniques to identify the most suitable classifier. The dataset used was the Afghanistan Demographic and Health Survey, and the study's results are expected to benefit the Afghanistan healthcare sector [1].

Similarly, in Ethiopia, a study focused on identifying determinants and developing a predictive model for skilled delivery service use. The study used logistic regression and machine-learning techniques, analyzing data from the 2016 Ethiopian Demographic and Health Survey. Several determinants were identified, including first antenatal care, birth order, television ownership, contraceptive use, cost needed for healthcare, age at first birth, and age at first sex. The J48 model showed the best predictive accuracy, sensitivity, specificity, and area under the ROC. The authors suggested that the developed predictive model could help target interventions for pregnant women to ensure skilled assistance during childbirth. They also proposed the creation of a web-based application based on the study's results [2].

According to a 2016 report by the World Health Organization (WHO), skilled attendance at birth in Afghanistan has shown progress but with significant regional variations. The lowest coverage is observed in rural areas due to factors like poor infrastructure, limited availability of skilled birth attendants, and cultural barriers. The report recommends improving the quality of care provided by skilled birth attendants and addressing cultural barriers to increase coverage. The conclusion emphasizes the need for significant investments in increasing the number of skilled birth attendants and improving the coverage and quality of skilled attendance at birth in Afghanistan [3].

The Ministry of Public Health (MoPH) of Afghanistan has prioritized reducing maternal and child deaths. Efforts have been made through the Basic Package of Health Services (BPHS) and Essential Package of 11 Health Services (EPHS) frameworks, as stated in the Kabul Declaration for Maternal and Child Health

signed in 2015. While the rate of skilled care during childbirth has increased, it alone is not sufficient to reduce maternal and newborn mortality and morbidity. Assessing the quality of maternal and newborn health (MNH) services is crucial, considering national programs, facility readiness, health worker competencies, health worker-patient interactions, and the service environment. A large-scale facility assessment was conducted to evaluate the quality of routine MNH service provision in Afghanistan. The assessment provides data to guide efforts in ensuring high-quality care throughout pregnancy, childbirth, and the postnatal period [4].

In a study published in the Journal of Perinatology in 2019, researchers aimed to develop a machine learning model for predicting neonatal mortality in preterm infants. The study utilized electronic health records from the Children's Hospital of Philadelphia in the United States. They employed a random forest algorithm and evaluated the model's performance using various metrics. The results showed good discrimination and calibration, suggesting the model's potential as a tool to predict neonatal mortality and improve outcomes through early intervention [5].

Afghanistan faces significant challenges in maternal and child healthcare. According to a report by UNICEF and WHO, Afghanistan has the highest rank of maternal mortality globally. Skilled child delivery services are inadequate, with less than half of the births occurring in health facilities. Home deliveries in unhygienic environments are common, particularly in rural communities. Political instabilities, civilian wars, and cultural restrictions contribute to the lack of clinical services and trained healthcare providers, especially midwives. The use of computational tools, such as data mining techniques, can extract valuable information and knowledge from available datasets. Efforts should be made to equip existing health professionals and efficiently deploy skilled child delivery services. The high infant and under-five mortality rates in Afghanistan highlight the challenges in providing adequate healthcare services, especially in rural and remote areas [6].

Access to skilled birth attendants is crucial for reducing maternal and child mortality. Skilled birth attendants, such as doctors, midwives, and nurses, have the necessary training and expertise to handle childbirth complications and provide essential care to both the mother and newborn. Increasing the availability and accessibility of skilled birth attendants is a key priority in improving maternal and child health outcomes. [7]

Adequate antenatal care plays a significant role in ensuring safer childbirth. Regular antenatal check-ups allow healthcare professionals to monitor the health of the mother and identify any potential risks or complications early on. Antenatal care visits provide an opportunity for education, counseling, and preparation for childbirth, including information on the importance of skilled assistance during delivery. [8]

Community-based interventions can contribute to improving access to skilled child delivery services, particularly in remote or underserved areas. These interventions may involve training and deploying skilled birth attendants at the community level, establishing maternity waiting homes near health facilities, or implementing transportation schemes to enable pregnant women to reach healthcare facilities in a timely manner. [9]

Ensuring the quality of skilled child delivery services is equally important as increasing their coverage. Quality improvement initiatives may involve ongoing training and capacity building for healthcare providers, implementing evidence-based practices, promoting respectful maternity care, and strengthening health systems to support safe and effective childbirth. [10]

Addressing sociocultural factors and empowering women can positively impact the utilization of skilled child delivery services. Efforts should be made to raise awareness about the importance of skilled assistance during childbirth and overcome barriers related to cultural practices, gender norms, and healthcare-seeking behaviors. Engaging communities, religious leaders, and local influencers can help foster a supportive environment for skilled child delivery. [11]

Chapter 3: Research Methodology

3.1 Data Gathering

The dataset used in this project was collected from Rabia Balkhi hospital. The data was obtained from illness documents, and I manually entered it into the dataset. The dataset contains 1969 rows and 10 columns (or features): Age, number of previous deliveries, number of miscarriages or stillbirths, Diabetes, Hypertension, Cesarean section, Normal delivery, Gestational Age, Mother Alive, and Baby Alive. Eight of these features are independent variables, which were obtained from the mother before delivery. The remaining two features, Mother Alive and Baby Alive, are dependent variables and are the result of the delivery. I consulted with Doctor Shaima Sadiqi, who specializes in finding the causes of mother deaths during delivery and how to prevent them, and Doctor Shagofa, who is skilled in delivery at Rabia Balkhi hospital, to choose these features. I divided the dataset into two parts, one for the mother and one for the baby, and I plan to create separate models for each.

Out[16]:

	Mother Age	Number of Previous Pregnancies	Number of Miscarriage or stillbirth	Diabetes	Hypertension	Cesarean Section	Normal Delivery	Gestational Age	Mother alive	Baby alive
0	23	3	0	0	1	0	1	38	1	1
1	30	3	0	0	0	0	1	38	1	1
2	30	5	1	0	0	0	1	38	1	1
3	27	3	0	0	0	0	1	38	1	1
4	23	5	3	0	0	0	1	38	1	1

Figure [2] General dataset

Out[2]:

	Mother Age	Number of Previous Pregnancies	Number of Miscarriage or stillbirth	Diabetes	Hypertension	Cesarean Section	Normal Delivery	Gestational Age	Mother alive
0	23	3	0	0	1	0	1	38	1
1	30	3	0	0	0	0	1	38	1
2	30	5	1	0	0	0	1	38	1
3	27	3	0	0	0	0	1	38	1
4	23	5	3	0	0	0	1	38	1

Figure [3] Separated dataset

3.2 Preprocessing Steps

Before training the machine learning model, a preprocessing step was performed on the dataset to ensure the data was appropriately prepared for analysis. The following preprocessing techniques were applied to the dataset:

3.2.1 Handling Missing Values

Any instances with missing values were removed from the dataset to ensure the integrity of the data used for training and evaluation. This was achieved by dropping rows containing missing values using the `dropna()` function.

3.2.2 Data Scaling - Standardization

To enhance the performance and convergence of the machine learning algorithm, the input features were scaled using a technique called standardization. Standardization transforms the features such that they have zero mean and unit variance.

The standardization process involves the following steps

Calculate the mean (μ) and standard deviation (σ) of each feature in the training set. For each feature, subtract the mean and divide by the standard deviation. This centers the feature distribution around zero and scales it to have unit variance. The formula for standardization of a feature x is given by: $z = (x - \mu) / \sigma$, where z is the standardized value. The `StandardScaler` from the `scikit-learn` library was employed to perform the standardization of the features in this project.

Standardization ensures that the features are on a similar scale, removing any bias due to differing scales among the input features. This is particularly important for algorithms that are sensitive to the scale of the features, such as distance-based algorithms like K-Nearest Neighbors or Support Vector Machines.

Splitting the Dataset

The dataset was split into training and testing sets using the `train_test_split` function from the `scikit-learn` library. This splitting process allows for the evaluation of the model's performance on unseen data.

The dataset was divided into 80% for training and 20% for testing, ensuring an adequate amount of data for both training and evaluation purposes.

The preprocessing step was crucial in preparing the dataset for the subsequent training and evaluation phases. By handling missing values and performing standardization on the features, potential biases and discrepancies in the data were addressed. Furthermore, splitting the dataset into training and testing sets enabled the evaluation of the model's generalization capabilities.

These preprocessing techniques contribute to the overall robustness and reliability of the machine learning model, ensuring accurate predictions and reliable result.

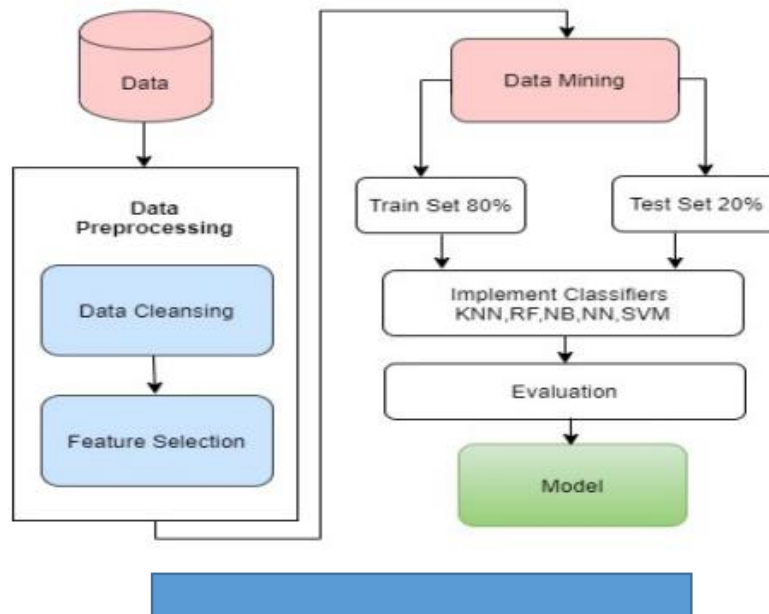


Figure [4] **Implementation of the predictive model**

3.3 Algorithm Selection and Evaluation

In this machine learning project, I utilized six different algorithms to train and evaluate my model. These algorithms were chosen based on their suitability for the task at hand and their potential to deliver accurate predictions. The algorithms used in this project are as follows:

3.3.1 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a popular machine learning algorithm used for both classification and regression tasks. It is a non-parametric and instance-based algorithm, meaning it doesn't make any assumptions about the underlying data distribution. Instead, it uses the available data directly to make predictions.

In KNN, the "K" refers to the number of nearest neighbors that are considered when making a prediction. When given a new input, the algorithm finds the K closest data points in the training set based on a distance metric, typically Euclidean distance. The predicted output is then determined by a majority vote (for classification) or the average of the nearest neighbors' values (for regression).

Here's a step-by-step overview of the KNN algorithm:

1. **Load the training data:** The algorithm begins by loading the labeled training data, which consists of input features and corresponding output labels.
2. **Choose the value of K:** Determine the number of nearest neighbors (K) to consider when making predictions. This value is typically chosen based on cross-validation or other model selection techniques.
3. **Calculate distances:** Compute the distance between the input data point and all other data points in the training set. Euclidean distance is a commonly used distance metric, but other metrics can also be employed.
4. **Select K nearest neighbors:** Identify the K data points with the smallest distances to the input data.
5. **Make predictions:** For classification tasks, assign the class label based on the majority vote of the K nearest neighbors. For regression tasks, take the average of the output values of the K nearest neighbors.
6. **Output the prediction:** Return the predicted class label or numerical value as the final output of the algorithm.

KNN has several strengths and weaknesses:

Strengths:

- Simple and easy to understand.
- Effective for multi-class classification tasks.
- Can handle both classification and regression problems.
- Doesn't make assumptions about the data distribution.

Weaknesses:

- Computationally expensive for large datasets.
- Sensitivity to the choice of K and the distance metric.
- Requires careful preprocessing of data and normalization.
- Doesn't work well with high-dimensional data.

It's important to note that KNN is a lazy learning algorithm, meaning it doesn't learn an explicit model during the training phase. Instead, it memorizes the training data and performs computations at prediction time. This characteristic makes KNN suitable for dynamic or non-stationary data where the underlying distribution may change over time. [12]

3.3.2 Support Vector Machines (SVM)

Support Vector Machines (SVMs) are a powerful supervised machine learning algorithm used for both classification and regression tasks. SVMs are particularly effective when dealing with complex datasets with a clear margin of separation between classes or when data is not linearly separable.

The key idea behind SVMs is to find an optimal hyperplane that maximally separates the data points of different classes. A hyperplane is a decision boundary that divides the feature space into different regions corresponding to different classes. SVMs aim to find the hyperplane with the largest margin, which is the maximum distance between the hyperplane and the nearest data points of any class. This margin maximization helps improve the generalization ability of the model.

Here are the main characteristics and steps involved in the SVM algorithm:

1. **Kernel trick:** SVMs utilize the kernel trick, allowing them to operate in high-dimensional feature spaces without explicitly calculating the transformed features. This allows SVMs to capture complex patterns and make nonlinear classifications by implicitly mapping the data into a higher-dimensional space.
2. **Margin optimization:** SVMs aim to find the hyperplane that maximizes the margin separating the classes. This hyperplane is selected by solving an optimization problem that minimizes the classification error while maximizing the margin. The data points that lie on the margin or violate the margin are called support vectors and play a crucial role in defining the decision boundary.
3. **Soft margin and regularization:** In cases where the data is not perfectly separable, SVMs can be extended to allow for some misclassifications. This is achieved through the use of a soft margin, which allows a certain number of data points to be misclassified. The trade-off between the margin size and the misclassification penalty is controlled by a regularization parameter (C) that balances the need for a larger margin with the desire for accurate classification.

4. **Kernel selection:** SVMs offer various kernel functions, such as linear, polynomial, radial basis function (RBF), and sigmoid, which define the similarity measure between data points. The choice of kernel depends on the data and the problem at hand. The RBF kernel is commonly used as it can handle complex, nonlinear relationships effectively.

5. **Model training and prediction:** To train an SVM model, labeled training data is used to find the optimal hyperplane. During prediction, the learned model is used to classify new, unseen data points by assigning them to one of the predefined classes based on their position relative to the decision boundary.

Some key advantages of SVMs include:

- Effective in high-dimensional and complex datasets.
- Robust against overfitting, thanks to the margin maximization principle.
- Versatility in handling both linearly separable and non-linearly separable data through kernel functions.
- Well-founded theoretical foundations and strong generalization capabilities.

SVMs also have a few considerations:

- Computationally expensive, especially for large datasets.
- Sensitivity to parameter tuning, such as the choice of the kernel and regularization parameter.
- Difficulty in interpreting the learned model compared to some other algorithms.

Overall, SVMs are widely used in various domains, including image classification, text categorization, bioinformatics, and finance, where they have demonstrated strong performance and robustness. [12]

3.3.3 Random Forest (RF)

The Random Forest algorithm is a popular ensemble learning method used for both classification and regression tasks in machine learning. It combines the predictions of multiple decision trees to make more accurate and robust predictions.

Here are the key features and steps involved in the Random Forest algorithm:

1. **Ensemble of decision trees:** Random Forest builds an ensemble of decision trees, where each tree is trained on a random subset of the training data. This process is known as bootstrapping or random

sampling with replacement. By using multiple trees, Random Forest reduces overfitting and improves generalization.

2. Random feature selection: During the construction of each decision tree, Random Forest further introduces randomness by considering only a subset of features at each split. This process helps to decorrelate the trees and promotes diversity among them. The number of features considered at each split is typically the square root or a logarithmic value of the total number of features.

3. Training the decision trees: Each decision tree in the Random Forest is trained using a modified version of the CART (Classification and Regression Trees) algorithm. The trees are built recursively by splitting the data based on certain features and thresholds to maximize information gain (for classification) or minimize impurity (for regression).

4. Voting for classification, averaging for regression: For classification tasks, the class prediction is determined by majority voting among the trees in the forest. Each tree "votes" for a class, and the class with the most votes becomes the final prediction. For regression tasks, the output values from all the trees are averaged to obtain the final prediction.

5. Out-of-bag (OOB) evaluation: Random Forest can estimate the performance of the model without the need for a separate validation set by utilizing the out-of-bag samples. Since each tree is trained on a different subset of the data, the samples that are not selected in the bootstrap process can be used for validation. This provides an unbiased estimate of the model's performance.

Some key advantages of the Random Forest algorithm include:

- Robustness against overfitting due to ensemble learning and random feature selection.
- Ability to handle high-dimensional data and large feature spaces.
- Provides estimates of feature importance, allowing for feature selection.
- Efficient for training and prediction, especially with large datasets.

However, it's important to consider a few limitations:

- Random Forests can be computationally expensive for training and require more memory compared to single decision trees.

- Interpretability of the model can be challenging due to the complexity of the ensemble.
- Random Forests may not perform well on datasets with strong linear relationships, as they tend to capture nonlinear patterns more effectively.

Random Forests are widely used in various domains, including finance, healthcare, and ecology, where they have shown excellent performance and versatility. They are particularly useful when dealing with complex datasets, handling missing values, and working with a mixture of categorical and numerical features. [12]

3.3.4 Neural Networks (NN)

Neural networks, also known as Artificial Neural Networks (ANNs), are a class of machine learning algorithms inspired by the structure and functioning of the human brain. Neural networks are widely used for various tasks, including classification, regression, pattern recognition, and more.

Here are the key features and steps involved in the Neural Network algorithm:

1. **Neurons and layers:** Neural networks consist of interconnected nodes called neurons organized into layers. The three main types of layers are the input layer, hidden layers, and output layer. The input layer receives the input data, while the output layer produces the final predictions. The hidden layers, which can be one or more, perform computations and extract features from the input.
2. **Neuron activation:** Each neuron in a neural network applies an activation function to the weighted sum of its inputs to introduce non-linearity into the model. Common activation functions include the sigmoid function, hyperbolic tangent (tanh) function, and rectified linear unit (ReLU) function. Activation functions allow neural networks to model complex relationships between inputs and outputs.
3. **Feedforward propagation:** In the feedforward step, the input data is passed through the network from the input layer to the output layer. The computations involve multiplying the input values with the corresponding weights, applying the activation function at each neuron, and passing the results to the next layer.
4. **Backpropagation and learning:** Backpropagation is the key step in training a neural network. It involves calculating the error or loss between the predicted output and the actual output, and then propagating this error backward through the network to adjust the weights. This adjustment is done using

optimization algorithms like gradient descent to minimize the error and improve the network's performance.

5. Training and optimization: Neural networks are trained by iteratively presenting training samples to the network, calculating the error, and updating the weights using backpropagation. The process continues for multiple epochs until the network learns the underlying patterns and achieves satisfactory performance. Optimization techniques such as learning rate adjustment, regularization, and early stopping are often used to improve training efficiency and prevent overfitting.

6. Model evaluation and prediction: Once the neural network is trained, it can be used for making predictions on new, unseen data. The input data is fed into the network, and the output layer provides the predicted values or class probabilities based on the learned patterns.

Some key advantages of neural networks include:

- Ability to learn complex nonlinear relationships between inputs and outputs.
- Capability to handle large amounts of data and high-dimensional feature spaces.
- Versatility in solving various machine learning tasks, including classification, regression, and more.
- Adaptability to different problem domains through the choice of network architecture and hyper parameters.

However, neural networks also have some considerations:

- Neural networks are computationally intensive and may require significant computational resources, especially for training large networks.
- They can be sensitive to the choice of hyper parameters, such as the number of layers, number of neurons per layer, and learning rate.
- Interpreting the learned representations and decision-making process of neural networks can be challenging.

Neural networks have achieved remarkable success in various domains, including computer vision, natural language processing, speech recognition, and many other fields. Different architectures, such as

Convolutional Neural Networks (CNNs) for image analysis and Recurrent Neural Networks (RNNs) for sequential data, have been developed to address specific problem types and improve performance. [12]

3.3.5 Logistic Regression

Logistic Regression is a popular statistical algorithm used for binary classification tasks. Despite its name, logistic regression is actually a classification algorithm rather than a regression algorithm. It models the relationship between a dependent variable and one or more independent variables by estimating the probabilities of the binary outcomes.

Here are the key features and steps involved in the Logistic Regression algorithm:

- 1. Sigmoid function:** Logistic Regression uses the sigmoid function (also known as the logistic function) to map the output of a linear function to a value between 0 and 1. The sigmoid function has an S-shaped curve, allowing it to represent the probability of an event occurring.
- 2. Hypothesis representation:** In logistic regression, the hypothesis function represents the relationship between the independent variables and the probability of the binary outcome. It is defined as $h\theta(x) = g(\theta^T * x)$, where $h\theta(x)$ is the predicted probability, $g()$ is the sigmoid function, θ is the vector of coefficients (weights), and x is the vector of independent variables.
- 3. Parameter estimation:** The goal of logistic regression is to estimate the optimal values of the coefficients (weights) θ that best fit the training data. This is typically done using maximum likelihood estimation or gradient descent optimization methods, where the objective is to minimize the difference between the predicted probabilities and the actual binary outcomes.
- 4. Decision boundary:** Logistic Regression uses a decision boundary to classify the data points into different classes based on the predicted probabilities. The decision boundary is a threshold value (e.g., 0.5) that separates the two classes. If the predicted probability is above the threshold, the data point is classified as one class, and if it's below, it's classified as the other class.
- 5. Regularization:** To prevent overfitting and improve generalization, logistic regression can incorporate regularization techniques such as L1 regularization (Lasso) or L2 regularization (Ridge). Regularization adds a penalty term to the cost function, reducing the impact of large coefficient values and promoting simpler models.

6. Model evaluation: Logistic regression models are evaluated using various metrics such as accuracy, precision, recall, F1 score, and area under the receiver operating characteristic (ROC) curve. These metrics provide insights into the model's performance in terms of correctly predicting the binary outcomes.

Some key advantages of logistic regression include:

- **Simplicity and interpretability:** Logistic regression provides a straightforward interpretation of the coefficients, indicating the direction and strength of the relationship between the features and the probability of the outcome.
- **Efficiency and scalability:** Logistic regression is computationally efficient and can handle large datasets with a relatively small number of features.
- **Robustness to noise:** Logistic regression is less sensitive to outliers and noise compared to some other classification algorithms.

However, there are a few considerations:

- Logistic regression assumes a linear relationship between the independent variables and the log-odds of the outcome. Non-linear relationships may require feature engineering or the use of other algorithms.
- It is primarily designed for binary classification tasks and may require modifications or extensions for multi-class classification.
- Logistic regression assumes independence of observations, and violations of this assumption may impact the model's performance.

Logistic regression is widely used in various domains, including healthcare, finance, marketing, and social sciences, where binary classification is commonly required. It serves as a valuable tool when interpretability, simplicity, and efficiency are desired. [12]

3.3.6 Naive Bayes (NB)

The Naive Bayes algorithm is a popular probabilistic classification algorithm based on Bayes' theorem. It is known as "naive" because it assumes that all features are independent of each other, which is a simplifying assumption that may not hold in real-world scenarios. Despite this assumption, Naive Bayes

has proven to be effective in many practical applications, especially in text classification and spam filtering.

Here are the key features and steps involved in the Naive Bayes algorithm:

1. **Bayes' theorem:** Naive Bayes is based on Bayes' theorem, which describes the probability of an event given prior knowledge. Bayes' theorem is formulated as $P(A|B) = (P(B|A) * P(A)) / P(B)$, where $P(A|B)$ is the posterior probability of event A given event B, $P(B|A)$ is the likelihood of event B given event A, $P(A)$ is the prior probability of event A, and $P(B)$ is the prior probability of event B.

2. **Feature independence assumption:** Naive Bayes assumes that all features in the dataset are independent of each other, given the class labels. This assumption simplifies the computation of the likelihood term in Bayes' theorem, as it allows the conditional probabilities of individual features to be multiplied together.

3. **Training phase:** During the training phase, Naive Bayes calculates the prior probabilities of each class label and the conditional probabilities of each feature given each class label. The prior probabilities represent the proportion of training samples belonging to each class, while the conditional probabilities estimate the likelihood of each feature value given a specific class.

4. **Classification phase:** In the classification phase, Naive Bayes computes the posterior probability of each class label given the input features using Bayes' theorem. The class label with the highest posterior probability is assigned as the predicted class for the input data point. The posterior probability is calculated by multiplying the prior probability of the class with the product of the conditional probabilities of the features given the class.

5. **Laplace smoothing:** To handle cases where a feature value in the test data has not been seen during training, Naive Bayes employs Laplace smoothing (also known as additive smoothing). Laplace smoothing adds a small constant to the numerator and adjusts the denominator accordingly to account for unseen feature values and prevent zero probabilities.

6. **Model evaluation:** Naive Bayes models are typically evaluated using metrics such as accuracy, precision, recall, F1 score, and area under the receiver operating characteristic (ROC) curve. These metrics assess the model's performance in correctly classifying instances.

Some key advantages of Naive Bayes include:

- **Simplicity and efficiency:** Naive Bayes is computationally efficient and requires a small amount of training data to estimate probabilities. It can handle high-dimensional datasets with a large number of features.
- **Good performance on text classification tasks:** Naive Bayes has been particularly successful in text-related tasks, such as spam detection, sentiment analysis, and document categorization.
- **Robustness to irrelevant features:** Naive Bayes can handle irrelevant features and is less affected by the curse of dimensionality.

However, there are a few considerations:

- **Independence assumption:** The assumption of feature independence may not hold in real-world scenarios, which can affect the accuracy of the model.
- **Sensitivity to feature distributions:** Naive Bayes assumes that the features follow a specific distribution (e.g., Gaussian, multinomial, or Bernoulli). If the data violates these assumptions, performance may be impacted.
- **Lack of model interpretability:** Naive Bayes does not provide insights into the relationships between features, as it assumes independence.

Naive Bayes is widely used in various domains, especially in text classification tasks where it has demonstrated strong performance. It serves as a fast and efficient algorithm for classification problems, particularly when the feature independence assumption is reasonable or when the algorithm is combined with feature selection techniques. [12]

To determine the most suitable algorithm for my project, I evaluated their performance using various metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into the models' accuracy and their ability to correctly identify positive instances.

After comprehensive evaluation, the Random Forest (RF) algorithm emerged as the optimal choice for my project. RF demonstrated high accuracy, precision, recall, and F1-score, making it well-suited for accurate predictions and reliable results.

3.4 Evaluation metrics for each algorithm

- Accuracy
- Precision
- Recall
- F1_score

3.4.1 Accuracy: Accuracy is a common metric used to evaluate the performance of classification models. It measures the proportion of correctly classified instances out of the total number of instances in a dataset. Accuracy provides a general overview of how well the model is performing in terms of overall correctness.

The formula to calculate accuracy is:

$$\text{Accuracy} = (\text{Number of correctly classified instances}) / (\text{Total number of instances})$$

Here are some key points to understand about accuracy:

1. **Interpretation:** Accuracy is typically expressed as a percentage, ranging from 0 to 100%. A higher accuracy value indicates a better-performing model with a higher proportion of correctly classified instances.
2. **Importance of class distribution:** Accuracy can be a useful metric when the class distribution in the dataset is balanced, meaning that the number of instances in each class is roughly equal. However, in imbalanced datasets where one class significantly outweighs the others, accuracy may not provide an accurate representation of the model's performance. In such cases, other metrics like precision, recall, or F1 score may be more informative.
3. **Limitations in imbalanced datasets:** In scenarios where the classes are imbalanced, a classifier that always predicts the majority class can achieve a high accuracy simply because it correctly predicts the majority class most of the time. This can be misleading, as the model may perform poorly

on the minority class. Therefore, it is important to consider additional metrics to gain a comprehensive understanding of model performance.

4. Accuracy as a standalone metric: While accuracy is a commonly used metric, it may not always provide a complete picture of a model's effectiveness, especially when the costs of false positives and false negatives are significantly different. In such cases, it is important to consider other metrics like precision, recall, F1 score, or area under the receiver operating characteristic (ROC) curve, which provide more insight into specific aspects of the model's performance.

5. Evaluation on test data: Accuracy is typically calculated on a separate test dataset that the model has not been trained on. This ensures that the accuracy metric represents the model's ability to generalize to new, unseen data.

6. Limitations: Accuracy alone does not provide information about the types of errors the model is making or the specific instances that are misclassified. For a more comprehensive evaluation, it is important to analyze other metrics, perform error analysis, and consider the context and requirements of the problem domain.

Overall, accuracy is a commonly used metric to assess the overall performance of classification models. However, it is important to consider other metrics and take into account the characteristics of the dataset and the specific goals of the problem when evaluating a model's effectiveness. [13]

3.4.2 Precision: Precision is a metric commonly used in binary classification tasks to evaluate the performance of a model. It measures the proportion of correctly predicted positive instances out of the total instances predicted as positive. Precision provides insights into the model's ability to avoid false positives.

The formula to calculate precision is:

$$\text{Precision} = (\text{True Positives}) / (\text{True Positives} + \text{False Positives})$$

Here are some important points to understand about precision:

1. **Interpretation:** Precision is typically expressed as a percentage, ranging from 0 to 100%. A higher precision value indicates a lower rate of false positives, meaning that the model is more accurate in identifying positive instances.

2. **Focus on positive predictions:** Precision focuses on the instances that the model predicts as positive. It measures how many of those predictions are correct.

3. **Importance of false positives:** Precision is particularly useful when false positives are costly or undesirable. For example, in medical diagnosis, incorrectly classifying a healthy person as having a disease (false positive) can lead to unnecessary treatments or anxiety. In such cases, achieving a high precision is crucial.

4. **Trade-off with recall:** Precision and recall are complementary metrics, often considered together. While precision measures the proportion of correctly predicted positive instances, recall measures the proportion of actual positive instances that are correctly identified by the model. There is often a trade-off between precision and recall, where improving one may lead to a degradation in the other.

5. **Evaluation on test data:** Precision is typically calculated on a separate test dataset that the model has not been trained on. This ensures that the precision metric represents the model's ability to generalize to new, unseen data.

6. **Limitations:** Precision alone does not provide information about the model's performance on negative instances or the instances that are incorrectly classified as negative (false negatives). Therefore, it is important to consider other metrics like recall, F1 score, or the specific requirements of the problem domain.

Precision is an important metric, particularly when the focus is on minimizing false positives. However, it should be considered alongside other metrics to obtain a comprehensive evaluation of the model's performance. The choice of evaluation metrics depends on the specific goals, class distribution, and costs associated with different types of classification errors in the problem domain.

[13]

3.4.3 Recall: Recall, also known as sensitivity or true positive rate, is a metric commonly used in binary classification tasks to evaluate the performance of a model. It measures the proportion of actual positive instances that are correctly identified by the model. Recall provides insights into the model's ability to avoid false negatives.

The formula to calculate recall is:

$$\text{Recall} = (\text{True Positives}) / (\text{True Positives} + \text{False Negatives})$$

Here are some important points to understand about recall:

1. **Interpretation:** Recall is typically expressed as a percentage, ranging from 0 to 100%. A higher recall value indicates a lower rate of false negatives, meaning that the model is more accurate in identifying positive instances.
2. **Focus on actual positive instances:** Recall measures the performance of the model in correctly identifying the positive instances from the total number of actual positive instances.
3. **Importance of false negatives:** Recall is particularly useful when false negatives are costly or undesirable. For example, in medical diagnosis, incorrectly classifying a person with a disease as healthy (false negative) can lead to delayed treatment or missed opportunities for intervention. In such cases, achieving a high recall is crucial.
4. **Trade-off with precision:** Recall and precision are complementary metrics that are often considered together. While recall measures the proportion of actual positive instances that are correctly identified, precision measures the proportion of positive predictions that are actually correct. There is often a trade-off between recall and precision, where improving one may lead to a degradation in the other.
5. **Evaluation on test data:** Recall is typically calculated on a separate test dataset that the model has not been trained on. This ensures that the recall metric represents the model's ability to generalize to new, unseen data.

6. **Limitations:** Recall alone does not provide information about the model's performance on negative instances or the instances that are incorrectly classified as negative (false positives). Therefore, it is important to consider other metrics like precision, F1 score, or the specific requirements of the problem domain.

Recall is an important metric, particularly when the focus is on minimizing false negatives and ensuring high sensitivity. However, it should be considered alongside other metrics to obtain a comprehensive evaluation of the model's performance. The choice of evaluation metrics depends on the specific goals, class distribution, and costs associated with different types of classification errors in the problem domain. [13]

3.4.4 F1-score: The F1 score is a metric commonly used in binary classification tasks to assess the overall performance of a model. It combines the precision and recall metrics into a single score, providing a balanced measure of the model's ability to correctly classify positive instances while minimizing false positives and false negatives.

The F1 score is calculated using the following formula:

$$\text{F1 score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

Here are some important points to understand about the F1 score:

1. **Interpretation:** The F1 score ranges from 0 to 1, where a value of 1 represents perfect precision and recall, indicating a highly accurate and reliable model. A higher F1 score indicates a better balance between precision and recall.

2. **Harmonic mean of precision and recall:** The F1 score is calculated as the harmonic mean of precision and recall. The harmonic mean gives more weight to lower values, making the F1 score sensitive to both precision and recall. This means that the F1 score is lower when either precision or recall is low, providing a more balanced assessment of the model's performance.

3. **Trade-off between precision and recall:** The F1 score is particularly useful when there is a trade-off between precision and recall, as it considers both metrics simultaneously. It rewards models that

achieve high precision and recall simultaneously and penalizes models that have a significant imbalance between the two.

4. Evaluation on test data: The F1 score is typically calculated on a separate test dataset that the model has not been trained on. This ensures that the F1 score represents the model's ability to generalize to new, unseen data.

5. Limitations: While the F1 score provides a balanced evaluation of a model's performance, it may not be the ideal metric in all scenarios. For example, in cases where precision or recall is of greater importance, other metrics like precision or recall alone may be more informative. The choice of evaluation metric depends on the specific goals and requirements of the problem domain.

The F1 score is a widely used metric in binary classification tasks, especially when there is a need to balance precision and recall. It provides a comprehensive assessment of the model's performance by considering both false positives and false negatives. [13]

K-Nearest Neighbors (KNN):

Accuracy: 0.853	Precision: 0.922	Recall: 0.911	F1-score: 0.917
------------------------	-------------------------	----------------------	------------------------

Naive Bayes (NB):

Accuracy: 0.853	Precision: 0.910	Recall: 0.894	F1-score: 0.902
------------------------	-------------------------	----------------------	------------------------

Support Vector Machines (SVM):

Accuracy: 0.888	Precision: 0.888	Recall: 1.000	F1-score: 0.941
------------------------	-------------------------	----------------------	------------------------

Neural Networks (NN):

Accuracy: 0.898	Precision: 0.912	Recall: 0.980	F1-score: 0.945
------------------------	-------------------------	----------------------	------------------------

Logistic Regression:

Accuracy: 0.896	Precision: 0.906	Recall: 0.986	F1-score: 0.944
------------------------	-------------------------	----------------------	------------------------

Random Forest (RF):

Accuracy: 0.934	Precision: 0.948	Recall: 0.980	F1-score: 0.963
------------------------	-------------------------	----------------------	------------------------

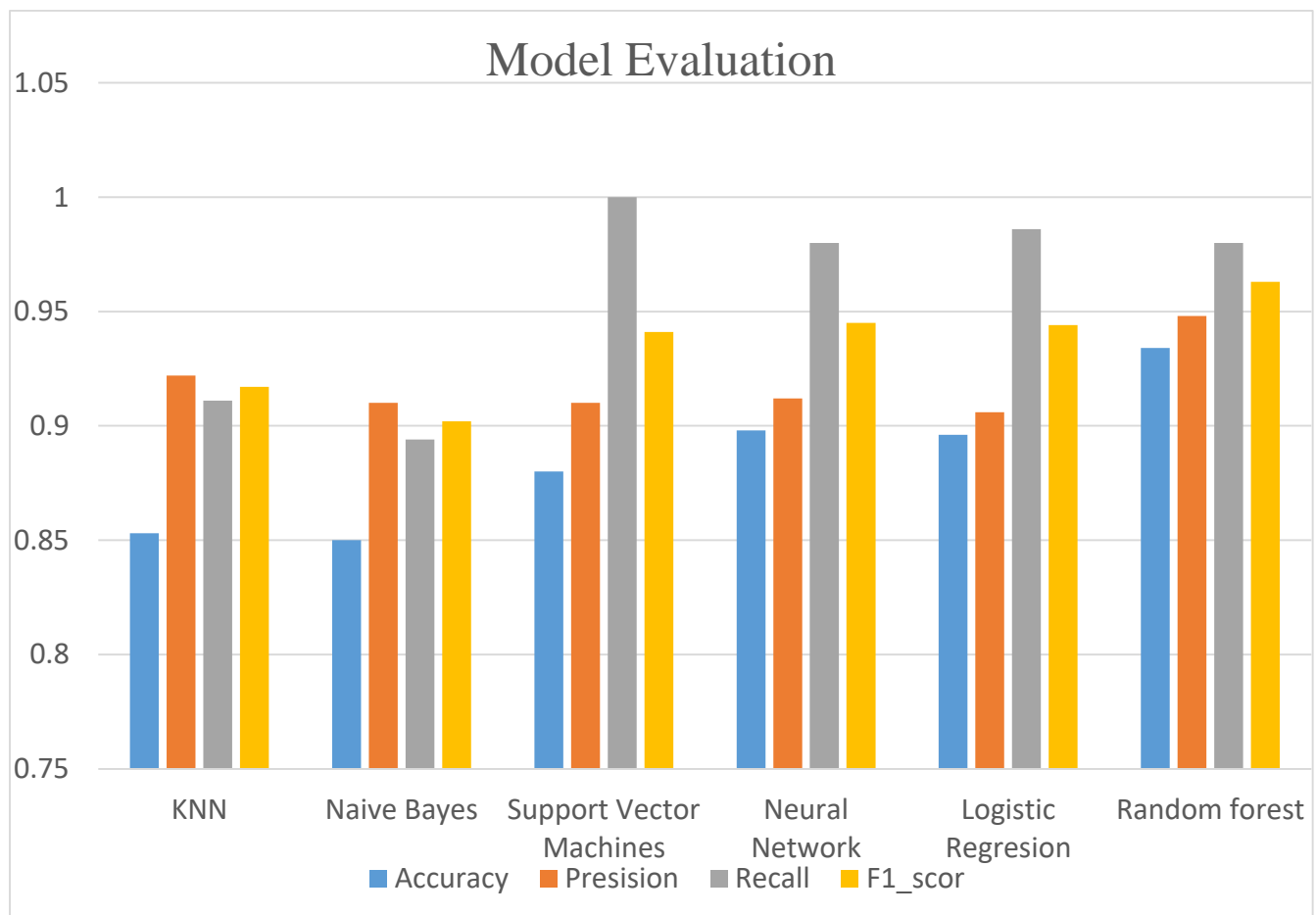


Figure [5] **Model Evaluation**

In the subsequent documentation, I will provide detailed insights into the model training processes, and more, focusing on the methodology and outcomes achieved using the Random Forest algorithm.

3.4 Selected Model

After performing the preprocessing step on the dataset, and algorithm selection, the following steps were carried out to train and evaluate the machine learning model:

Training the Random Forest Classifier

A Random Forest classifier was created using the scikit-learn library's RandomForestClassifier. The classifier was configured with 1000 trees (n_estimators=1000) and a random state of 42 (random_state=42).

The preprocessed training data was used to train the classifier by calling the fit () method on the Random Forest classifier object.

Making Predictions on the Testing Set

Predictions were made on the preprocessed testing data using the trained Random Forest classifier. The predict () method was used, passing the testing input features (X_test), which returned the predicted labels for the corresponding instances. The predicted labels were stored in the y_pred variable.

Evaluating the Model's Performance

The performance of the trained model was assessed using various evaluation metrics calculated based on the predicted labels (y_pred) and the actual labels from the testing set (y_test).

The following evaluation metrics were computed:

Accuracy: The proportion of correctly predicted instances to the total number of instances in the testing set. [13]

Precision: The ability of the model to correctly identify positive instances out of all instances predicted as positive. [13]

Recall: The ability of the model to correctly identify positive instances out of all actual positive instances. [13]

F1-score: The harmonic mean of precision and recall, providing a balanced measure of the model's performance. [13]

3.5 Making Predictions on New Data

The trained model can be used to make predictions on new, unseen data.

A predict_emergency_situation () function was implemented, which takes input values representing various features related to a specific scenario. The function uses the trained Random Forest classifier to classify the situation as either an "Emergency situation" or "No Emergency situation". An example usage is provided, demonstrating how to call the function with a set of input values and obtain the corresponding prediction. These steps, performed after the preprocessing stage, are crucial for training the model,

evaluating its performance, and making predictions on new data. The evaluation metrics provide insights into the model's accuracy, precision, recall, and F1-score, enabling an assessment of its effectiveness in real-world scenarios.

Chapter 4: Implementation Details

The implementation of the Emergency Situation Predictor involves the following steps:

4.1 Dataset Loading: The maternal health records dataset is loaded into the program using the `pandas` library's `read_csv()` function. This step ensures that the necessary data is available for training the model.

```
In [2]: # Load the dataset
dataset = pd.read_csv('D:/projec final/My Project Dataset_Mother.csv')
dataset.head()
```

Out[2]:

	Mother Age	Number of Previous Pregnancies	Number of Miscarriage or stillbirth	Diabetes	Hypertension	Cesarean Section	Normal Delivery	Gestational Age	Mother alive
0	23	3	0	0	1	0	1	38	1
1	30	3	0	0	0	0	1	38	1
2	30	5	1	0	0	0	1	38	1
3	27	3	0	0	0	0	1	38	1
4	23	5	3	0	0	0	1	38	1

Figure [6] Dataset Loading

Loading the Dataset:

The code loads a CSV dataset file named 'My Project Dataset_Mother.csv' using `pd.read_csv()` from pandas library and assigns it to the variable `dataset`.

The `head()` function is called to display the first few rows of the dataset.

4.2 Data Preprocessing: The dataset is preprocessed to handle missing values. Any rows or columns with missing values can be either removed or imputed with appropriate values depending on the specific requirements. In this implementation, the `dropna()` method from the `pandas` library is used to remove any rows with missing values.

```
[3]: # Handle missing values
data = dataset.dropna()

[4]: # Split the dataset into input features and labels
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
```

Figure [7] Data Preprocessing and handling missing values

4.2.1 Handling Missing Values:

The code drops rows with missing values using `dropna()` from pandas, and the resulting dataset without missing values is stored in the variable `data`.

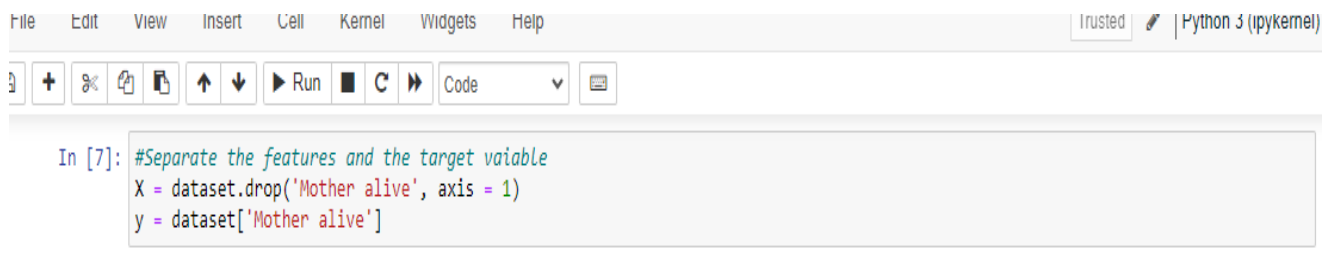
4.2.2 Splitting the Dataset:

The dataset is split into input features and labels.

The input features are extracted from the dataset using `iloc[:, :-1].values`, which selects all rows and all columns except the last column. They are stored in the variable `X`.

The labels are extracted using `iloc[:, -1].values`, which selects all rows and the last column. They are stored in the variable `y`.


4.3 Feature Extraction: The dataset is split into input features (`X`) and labels (`y`). The input features are selected from the dataset, excluding the target variable. The labels represent the target variable, which in this case is whether an emergency situation occurs during childbirth.

A screenshot of a Jupyter Notebook interface. The top menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. On the right, it says 'Trusted' and 'Python 3 (ipykernel)'. Below the menu is a toolbar with icons for adding cells, undo, redo, and running code. The main area shows a code cell with the following text:

```
In [7]: #Separate the features and the target variable
X = dataset.drop('Mother alive', axis = 1)
y = dataset['Mother alive']
```

Figure [8] **Feature Extraction**

4.4 Data Scaling: The input features are standardized using the `StandardScaler` class from the `scikit-learn` library. Standardization ensures that all input features have zero mean and unit variance, which is important for many machine learning algorithms to perform well.

A screenshot of a Jupyter Notebook interface showing a code cell with the following text:

```
In [6]: # Scale the features using standardization
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Figure [9] Data Scaling and standardization

Feature Scaling:

The input features in the training set (`X_train`) and testing set (`X_test`) are standardized using `StandardScaler()` from `scikit-learn` to ensure all features have the same scale and are centered around zero.

4.5 Model Training: The Random Forest classifier is utilized as the machine learning model for the Emergency Situation Predictor. The `RandomForestClassifier` class from the `scikit-learn` library is used to create an instance of the classifier. The model is then trained on the preprocessed and scaled input features (X) and labels (y).

```
In [9]: # Create a Random Forest classifier with 1000 trees
rf_classifier = RandomForestClassifier(n_estimators=1000, random_state=42)

# Train the classifier
rf_classifier.fit(X_train, y_train)

# Make predictions on the testing set
y_pred = rf_classifier.predict(X_test)
```

Figure [10] Training RF model

Creating and Training the Random Forest Classifier:

- A random forest classifier is created with `RandomForestClassifier()` from scikit-learn. It is initialized with 1000 trees and a random state of 42.
- The `fit()` method is called on the random forest classifier to train it using the standardized input features (`X_train`) and corresponding labels (`y_train`).

4.6 Prediction Function: The `predict_emergency_situation()` function is defined to make predictions based on new input data. This function takes the input features as arguments, applies the necessary preprocessing steps (scaling), and uses the trained Random Forest classifier to predict whether an emergency situation is likely to occur during childbirth.

```
In [11]: def predict_emergency_situation(mother_age, num_prev_pregnancies, num_miscarriage_stillbirth, diabetes, hypertension, cesarean_section,
input_data = np.array([mother_age, num_prev_pregnancies, num_miscarriage_stillbirth, diabetes, hypertension, cesarean_section])
prediction = rf_classifier.predict(input_data)[0]

if prediction == 1:
    return 'Emergency situation'
return 'No Emergency situation'
```

Figure [11] Predict function

Defining the Prediction Function:

The `predict_emergency_situation()` function takes multiple input parameters related to a mother's pregnancy and health information.

- The input parameters are used to create an input data array, which is then passed to the trained random forest classifier for prediction.
- The prediction is stored in the variable `prediction`, and based on its value, the function returns either 'Emergency situation' or 'No Emergency situation'.

- The implementation of the Emergency Situation Predictor is modular and can be integrated into different applications or systems based on specific requirements. The code provided can serve as a starting point, and modifications can be made to suit the specific needs of the project.

It is important to note that the implementation assumes the availability of a suitable dataset with accurate and relevant maternal health records. Additionally, proper validation and evaluation should be performed to assess the accuracy and effectiveness of the predictor in the specific context of use.

4.7 Interface Development

A simple web-based interface was created to allow users to input relevant information and obtain predictions. The interface was developed using Flask, a Python web framework, which facilitated the integration of the trained Random Forest model with the interface.

Emergency Situation Predictor

Mother Age (years):

Number of Previous Pregnancies:

Number of Miscarriage or Stillbirth:

Diabetes (1 for Yes, 0 for No):

Hypertension (1 for Yes, 0 for No):

Cesarean Section (1 for Yes, 0 for No):

Normal Delivery (1 for Yes, 0 for No):

Gestational Age (weeks):

Figure [12] **Interface**

4.8 To use the Emergency Situation Prediction system

Access the web-based interface provided for the project. Enter the required information, such as the mother's age, number of previous pregnancies, history of miscarriages or stillbirths, presence of diabetes and hypertension, history of cesarean section and normal delivery, and gestational age.

4.8.1 Submit the form to initiate the prediction process: The system will utilize the trained Random Forest model to analyze the input data and provide a prediction regarding the likelihood of an emergency situation during pregnancy.

The result of the prediction will be displayed on the interface, indicating whether an emergency situation is likely or not.

4.9 The project code consists of the following components

4.9.1 app.py: The main Python script that handles the Flask application setup, routes, and prediction logic.

4.9.2 Main_page.html: The HTML template for the main page interface, where users input the required information.

4.9.3 result.html: The HTML template for the result page, where the predicted result is displayed. The app.py script loads the dataset, performs preprocessing, trains the Random Forest classifier, and defines the necessary routes for the application.

The **predict_emergency_situation** function is responsible for utilizing the trained model to make predictions based on the provided input features.

4.10 Getting Started

To run the Emergency Situation Prediction project locally, follow these steps:

Ensure you have Python installed on your machine. Install the required libraries by running `pip install flask, pandas, numpy, and scikit-learn` in your command line. Place the dataset file (Dataset_Mother.csv) in the same directory as the app.py script. Update the file paths in the code to match the actual location of the dataset and HTML files. Run the app.py script using the command `python app.py`. Access the application in your web browser at <http://localhost:5000>.

The Emergency Situation Predictor is implemented using Python programming language and relies on the following libraries:

pandas: A powerful data manipulation library used for loading and preprocessing the dataset.

numpy: A fundamental package for scientific computing with Python, used for numerical operations and array manipulation.

scikit-learn: A machine learning library that provides tools for data preprocessing, model training, and evaluation.

Flask: Flask is a popular web framework for building web applications and APIs using the Python programming language. It is known for its simplicity, flexibility, and ease of use. Flask is categorized as a micro framework because it provides only the essential features necessary to build web applications, allowing developers to have more control and flexibility in designing their applications.

Here are some key features and concepts related to Flask:

1. **Routing:** Flask uses a decorator-based approach to define routes, which map URLs to specific Python functions called view functions. These view functions are executed when a request matching the specified URL is received. Routing allows you to define different endpoints for handling various HTTP methods (GET, POST, etc.) and parameters.
2. **Templating:** Flask supports templating using Jinja2, a powerful and flexible template engine. Templating allows you to separate the presentation logic from the application logic, making it easier to generate dynamic HTML pages by embedding Python code within HTML templates.
3. **HTTP Request Handling:** Flask provides convenient methods to handle HTTP requests and access request data, such as form data, query parameters, and request headers. It also supports file uploads and handling cookies.
4. **Response Handling:** Flask allows you to return various types of responses, such as HTML, JSON, or files, from your view functions. It provides methods to set response headers, handle redirects, and handle error pages.
5. **Flask Extensions:** Flask has a rich ecosystem of extensions that provide additional functionality, such as database integration (e.g., Flask-SQLAlchemy, Flask-MongoEngine), authentication (e.g., Flask-Login, Flask-JWT), form handling (e.g., Flask-WTF), and more. These extensions can be easily integrated into Flask applications to extend their capabilities.
6. **Development Server:** Flask comes with a built-in development server, which is convenient for local development and testing. It automatically reloads the application when changes are detected, making the development process more efficient.
7. **Deployment:** Flask applications can be deployed using various web servers, such as Gunicorn or uWSGI, or as standalone applications using tools like Docker. Flask also supports integration with popular cloud platforms and hosting providers.

Flask's simplicity and minimalistic design make it a popular choice for small to medium-sized web applications and APIs. It provides a solid foundation for building web applications while allowing developers to make choices and customize their applications based on specific requirements. Flask's extensive documentation and active community make it easy to find resources and support when working with the framework. [14]

Chapter 5: Results

The final documentation provides an overview of a project focused on predicting emergency situations during childbirth and improving maternal and infant healthcare outcomes in Afghanistan. The study utilizes machine learning techniques and evaluates six different algorithms: K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forest (RF), Neural Networks (NN), Logistic Regression, and Naive Bayes. After comprehensive evaluation, the Random Forest algorithm is identified as the optimal choice due to its high accuracy and reliable results.

The proposed solution involves deploying the predictive model in both urban and rural settings. In urban areas, the model can be utilized in hospitals to predict whether a mother will have a normal delivery or require emergency interventions. This enables healthcare providers to take appropriate actions and provide timely care. Additionally, pregnant mothers at or beyond 30 weeks' gestation can input their information for stage-specific predictions.

In rural villages where access to hospitals and resources is limited, the predictive model becomes invaluable. Mothers can determine if an emergency situation is likely, allowing them to avoid unnecessary and risky journeys to distant hospitals. Instead, they can safely deliver with the assistance of midwives in their villages.

The documentation emphasizes the importance of leveraging computational tools and advanced analytics to reduce maternal and infant mortality rates in Afghanistan. By identifying high-risk cases through the predictive model, timely interventions can be made, and healthcare resources can be allocated more efficiently. The project's architecture, data preprocessing techniques, model training processes, and outcomes achieved using the Random Forest algorithm are thoroughly documented.

The results highlight the potential of machine learning in improving healthcare outcomes and addressing the unique challenges faced by Afghanistan in maternal and infant healthcare. The implementation of the predictive model can significantly contribute to reducing maternal and infant mortality rates in the country by providing early warnings and assisting healthcare providers in making informed decisions.

It is important to note that the study does not directly address the underlying socio-economic factors or structural challenges contributing to high maternal and infant mortality rates in Afghanistan. The focus of the project is primarily on the development and evaluation of the predictive model and its potential application within the given context.

Overall, the results and discussion presented in the final documentation demonstrate the effectiveness of the Random Forest algorithm in predicting emergency situations during childbirth and highlight the potential impact of implementing the predictive model in both urban and rural healthcare settings in Afghanistan.

5.1 Limitation

5.1.1 Limited availability of hospital data: The preparation of the dataset for this project poses a significant challenge in Afghanistan. Currently, there is no centralized collection of data in hospitals, making it difficult to obtain comprehensive and reliable data for analysis. The lack of readily available data hampers the ability to train and evaluate the predictive model effectively.

5.1.2 Challenges in data collection from patient documents: In the absence of structured hospital data, researchers may have to rely on manually collecting data from patient documents, such as medical records or birth certificates. This process can be time-consuming and prone to errors, as it requires extensive effort to extract relevant information and standardize the data for analysis.

5.1.3 Difficulty in data collection from women's hospitals: Due to cultural and regional constraints, it can be challenging for male researchers to access women's hospitals and collect data directly. The presence of Taliban government restrictions further complicates the situation, as men may not be allowed to enter hospitals where women deliver. This limitation adds an additional layer of complexity to the data collection process.

5.1.4 Ethical considerations: Given the sensitive nature of maternal and infant healthcare, it is essential to prioritize ethical considerations when collecting and analyzing data. Respecting patient privacy, ensuring informed consent, and safeguarding sensitive information should be fundamental principles in the data collection process. The limitations imposed by cultural norms and government restrictions may further complicate the ethical aspects of data collection and require careful navigation.

5.2 Future Work

5.2.1. Development of a mobile application: Building a mobile application based on the predictive model can greatly enhance its accessibility and usability. The application can provide a user-friendly interface for both healthcare providers and pregnant mothers to input relevant data and receive predictions and recommendations. The app can also include educational resources and information on prenatal care, emergency preparedness, and available healthcare services.

5.2.2 Offline functionality: Considering the limited internet access in many rural areas of Afghanistan, it would be beneficial to develop an offline version of the mobile application. This would enable users to access and utilize the predictive model even without an internet connection. An offline app would be particularly valuable for individuals living in remote villages where internet connectivity is scarce or unreliable.

5.2.3 Continuous data collection and model refinement: As data availability improves over time, continuous data collection efforts should be undertaken to enrich the dataset and enhance the predictive model's accuracy and generalizability. Long-term data collection initiatives can involve collaborations with hospitals, clinics, and community health workers to ensure comprehensive and representative data collection. Regular model updates and refinements based on the newly collected data can further improve the model's performance.

5.2.4 Evaluation in real-world settings: Once the predictive model and mobile application are developed, rigorous evaluation studies should be conducted in real-world healthcare settings across different regions of Afghanistan. These studies can assess the model's effectiveness, usability, and impact on maternal and infant health outcomes. Feedback from healthcare providers, pregnant mothers, and other stakeholders should be actively sought to iterate and improve the model based on their experiences and suggestions. By pursuing these avenues for future work, the predictive model can be further enhanced, and its reach and impact can be expanded to benefit a larger population, including those residing in remote villages with limited internet access. Continuous improvement, integration with existing systems, and real-world evaluation are crucial for the successful implementation and scalability of the model in Afghanistan's healthcare landscape.

Conclusion

In conclusion, this project addresses the critical healthcare challenges faced by Afghanistan, specifically focusing on the alarming rates of maternal and infant mortality and the limited access to skilled birth attendants and healthcare facilities. The aim of this project is to develop a predictive model using machine learning techniques that can effectively identify emergency situations during childbirth and provide early warnings to mitigate risks.

Through a meticulous evaluation of six machine learning algorithms, including K-Nearest Neighbors, Support Vector Machines, Random Forest, Neural Networks, Logistic Regression, and Naive Bayes, the Random Forest algorithm emerges as the optimal choice. With an impressive accuracy rate of 93% and consistent reliability, this algorithm demonstrates its potential to deliver accurate predictions.

The proposed solution offers a versatile approach that can be implemented in both urban and rural settings. In urban areas, the predictive model can be seamlessly integrated into hospital systems to assess whether a mother will require emergency interventions or experience a normal delivery. This empowers healthcare providers to take proactive measures and provide timely care. Additionally, pregnant mothers beyond 30 weeks' gestation can input their information to receive stage-specific predictions.

In rural villages with limited access to healthcare resources, the predictive model becomes an invaluable tool. It enables mothers to determine the likelihood of an emergency situation, allowing them to avoid risky journeys to distant hospitals. Instead, they can safely deliver under the guidance of local midwives within their own communities.

By leveraging the power of computational tools and advanced analytics, this research aims to make a substantial impact on reducing maternal and infant mortality rates in Afghanistan. The developed predictive model offers an innovative approach to identify high-risk cases, leading to prompt interventions and more efficient allocation of healthcare resources. The comprehensive documentation provided in this study delves into the project's architecture, data preprocessing techniques, model training processes, and the successful outcomes achieved using the Random Forest algorithm.

In summary, this project presents a promising solution that utilizes machine learning to enhance healthcare outcomes and address the unique challenges faced by Afghanistan's maternal and infant healthcare system. By accurately predicting emergency situations during childbirth, healthcare

professionals can intervene promptly, resulting in improved outcomes for both mothers and babies. This project contributes to the ongoing efforts to enhance healthcare in Afghanistan and serves as a solid foundation for future advancements in the field.

References

- [1] Nasratullah Nasrat, Application of Machine Learning in Assignment of Child Delivery Service in Afghanistan, May 2021, https://www.researchgate.net/publication/352533685_Application_of_Machine_Learning_in_Assignment_of_Child_Delivery_Service_in_Afghanistan
- [2] Suleman Atique, Tariq Azim, Migiretu M. Kebede, Predicting skilled delivery service use in Ethiopia, 05 November 2019, <https://bmcmmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-0942-5>
- [3] World Health Organization, Skilled delivery in Afghanistan: Challenges and opportunities, 2016 https://www.who.int/maternal_child_adolescent/documents/skilled-deliveryafghanistan/en/
- [4] Jhpiego Corporation, Afghanistan National Maternal and Newborn Health Quality of Care Assessment 2016, JAN 2017, <https://www.unicef.org/afghanistan/media/1806/file/afg-report-MNHQoC2016.pdf.pdf>
- [5] Katheria AC, Trivedi S, Gugino SF, Development and validation of a machine learning model for prediction of neonatal mortality in preterm infants, 2019 <https://www.nature.com/articles/s41372-018-0289-9>
- [6] World Bank (2021), Mortality rate, infant (per 1,000 live births) - Afghanistan. 2021. <https://data.worldbank.org/indicator/SP.DYN.IMRT.IN?locations=AF>
- [7] Access to skilled birth attendants is crucial for reducing maternal and child mortality <https://bmcpregnancychildbirth.biomedcentral.com/articles/10.1186/s12884-021-04290-7#Bib1>
- [8] Associations of socioeconomic determinants with community clinic awareness and visitation among women: evidence from Bangladesh Demographic and Health Survey-2011 <https://bmccresnotes.biomedcentral.com/articles/10.1186/s13104-015-1374-7>
- [9] St John's Mother & Baby Programme Supports Pregnant Women, Mothers & Infants. St John is Providing Urgent Healthcare for Women and Children to Prevent Mortality https://www.stjohninternational.org/Appeal/motherandbabyappeal?gclid=Cj0KCQiAsburBhCIARIsAExmsu4NIGLfrXWJrv4Nk2sCv48sViXJzBjAkmWVulGkM1bdDhLR2wVLeV8aAi8NEALw_wcB
- [10] P. Shah, "An introduction to weka," Opensourceforu.com, 10-Jan2017. [Online]. Available: <https://opensourceforu.com/2017/01/anintroduction-to-weka/> . [Accessed: 20-Oct-2020].
- [11] M. A. Hall, "Correlation-based feature selection for machine learning," Waikato.ac.nz. [Online]. Available: <https://www.cs.waikato.ac.nz/~ml/publications/1999/99MHThesis.pdf> . [Accessed: 04-Sep-2020]
- [12] JavaTpoint Website <https://www.javatpoint.com/>

- [13] Smart Vision website <https://www.sv-europe.com/crisp-dm-methodology/>
- [14] flask information <https://pythonbasics.org/what-is-flask-python/>

