Machine Learning Project Report

Maternal Health Risk

Student list:

Full name	Group	Section
KHEZZANE DINA	**04**	**02**
OUARI Meriem	**05**	**02**
LITIM CHIRAZ	**05**	**02**

Abstract

This project focuses on utilizing machine learning techniques to predict maternal health risks, leveraging a dataset collected from various healthcare facilities in rural Bangladesh. The dataset, titled "Maternal Health Risk," encompasses critical attributes , all of which are significant risk factors for maternal mortality. The data collection process was facilitated through an IoT-based risk monitoring system, ensuring real-time and comprehensive data acquisition.

The objective of this project is to develop a predictive model that can classify the risk intensity levels during pregnancy, aiding healthcare providers in early identification and intervention for high-risk cases. By overcoming the challenges associated with traditional risk assessment methods and leveraging advanced machine learning algorithms, this project aims to contribute to improving maternal health outcomes, aligning with the UN's Sustainable Development Goals (SDGs) regarding maternal health.

Through detailed analysis of the dataset's origin, significant attributes, summary statistics, potential visualizations and modeling, this project seeks to provide valuable insights into maternal health risk prediction, empowering healthcare professionals with actionable tools for better decision-making and intervention strategies.

Introduction

Maternal health is a critical aspect of public health, particularly in rural areas where access to healthcare services may be limited. The mortality rate among pregnant women remains a significant concern globally, with various risk factors contributing to adverse outcomes. Many pregnant women face life-threatening complications during pregnancy and post-pregnancy due to a lack of sufficient information about maternal health care. This gap in knowledge can lead to preventable deaths among pregnant women and neonates, highlighting the critical need for comprehensive monitoring and early intervention strategies.

In this context, machine learning offers promising opportunities to improve risk prediction and intervention strategies, ultimately enhancing maternal health outcomes. The machine learning project discussed in this report focuses on predicting maternal health risks using a dataset collected from diverse healthcare facilities in rural Bangladesh. The dataset encompasses essential attributes such as age, blood pressure, blood sugar levels, body temperature, heart rate, and risk levels, all of which are known risk factors for maternal mortality.

By leveraging advanced machine learning algorithms, this project aims to develop a predictive model that can classify the risk intensity levels during pregnancy, thereby aiding healthcare providers in early identification and intervention for high-risk cases. The problem addressed by this project is multifaceted. Firstly, it aims to overcome the challenges associated with traditional risk assessment methods, which may lack precision and timely intervention capabilities. Secondly, it seeks to address the disparities in maternal health outcomes, especially in underserved rural areas where access to specialized care may be limited. Thirdly, by leveraging IoT-based data collection systems, this project bridges the gap between data acquisition and actionable insights, empowering healthcare professionals with real-time risk assessment tools.

Through continuous monitoring and proactive intervention based on data-driven insights, we aim to reduce the preventable death toll among pregnant women and neonates, ultimately contributing to improved maternal health outcomes. This project aligns with the UN's Sustainable Development Goals (SDGs) and promotes equity in healthcare access for all pregnant women, regardless of their geographical location or socioeconomic status .

Dataset Description

The dataset utilized in this project, titled "Maternal Health Risk," originates from data collected across various healthcare facilities in rural Bangladesh. The data collection process was facilitated through an IoT-based risk monitoring system, ensuring real-time and comprehensive data acquisition. This dataset is crucial for understanding and predicting maternal health risks, with significant attributes contributing to risk assessment and intervention strategies.

Significant Attributes:

Variable Name	Role	Type	Description	Missing Values
Age	Feature	Integer	Any age in years when a woman is pregnant.	no
SystolicBP	Feature	Integer	Upper value of Blood Pressure in mmHg, another significant attribute during pregnancy.	no
DiastolicBP	Feature	Integer	Lower value of Blood Pressure in mmHg, another significant attribute during pregnancy.	no
BS	Feature	float	Blood glucose levels is in terms of a molar concentration	no
BodyTemp	Feature	float	Represents the body temperature, although specific units are not provided.	no
HeartRate	Feature	float	A normal resting heart rate	no
RiskLevel	Target	Categorical	Predicted Risk Intensity Level during pregnancy considering the previous attribute.	no

These attributes are integral to risk prediction models, as they provide essential insights into the health status of pregnant women and their potential risk of adverse outcomes.

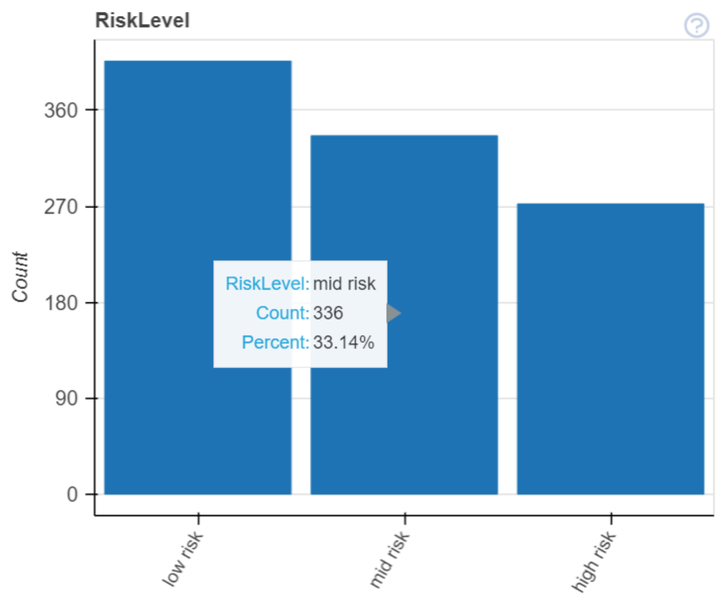
Description and statistical distribution of the dataset features.:

	Age	SystolicBP	DiastolicBP	BS	BodyTemp	RiskLevel
count	1014	1014	1014	1014	1014	1014
mean	29.8718	113.198	76.4606	8.72599	98.6651	74.3018
std	13.4744	18.4039	13.8858	3.29353	1.37138	8.0887
Min	10	70	49	6	98	7
25%	19	100	65	6.9	98	70

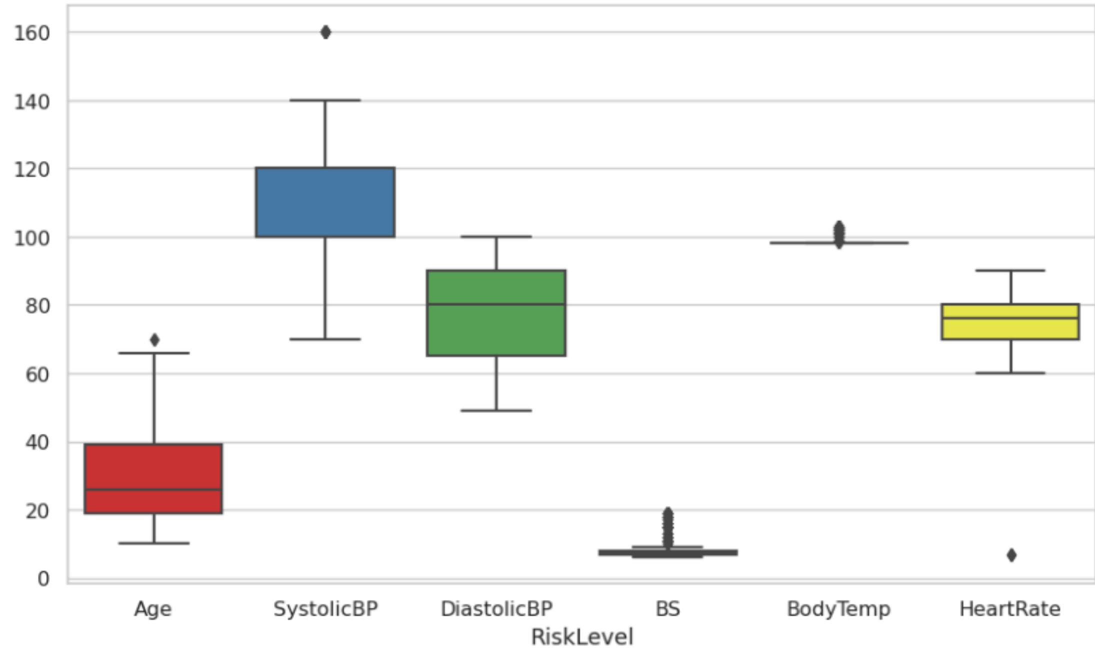
50%	26	120	80	7.5	98	76
75%	39	120	90	8	98	80
Max	70	160	100	19	103	90

Visualization:

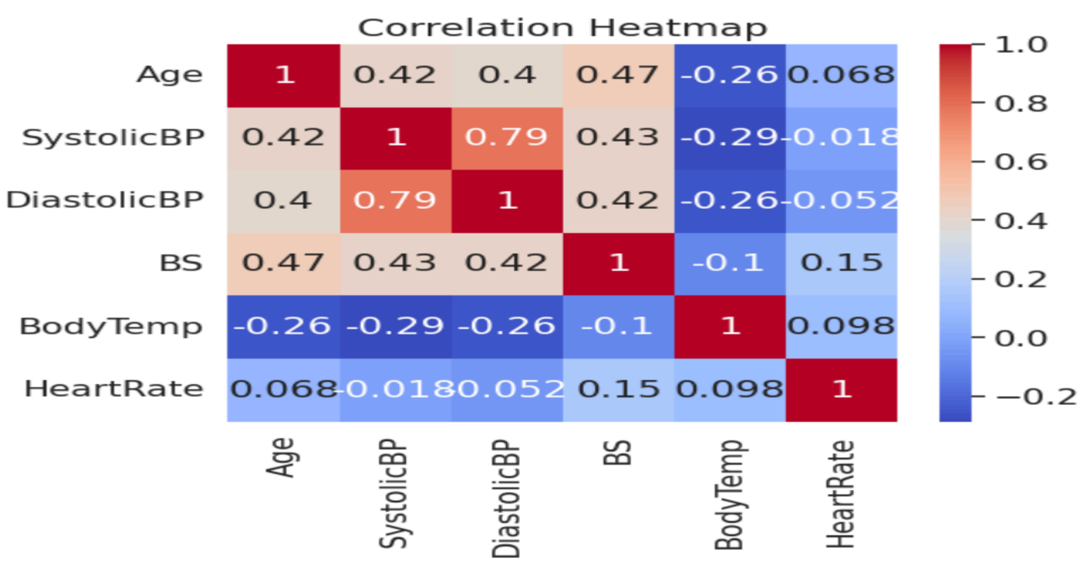
The dataset included a categorical target variable called Risk Level, which had three distinct classes: high risk, mid risk, and low risk. This contributed to a total of 7 variables in the dataset. Across the 1,014 instances in the dataset, there were 406 samples classified as low risk, 336 samples classified as mid risk, and 272 samples classified as high risk, as depicted in Figure 1.



Most variables exhibit outliers, which could skew their distribution. Notably, HeartRate stands out with an outlier that is significantly distant from the other values. Additionally, variables BS (Blood Sugar) and BodyTemp appear dense, indicating a high concentration of data within a narrow range. While this concentration is common and often normal, it's prudent to examine these variables further for any potential anomalies or data issues.



The variables SystolicBP and DiastolicBP exhibit a high correlation, with a correlation coefficient of 0.79 as depicted in the graph. This indicates that they contain highly similar information and show little to no variance in terms of information content. Therefore, removing one of these similar features should not pose an issue, given their high correlation .



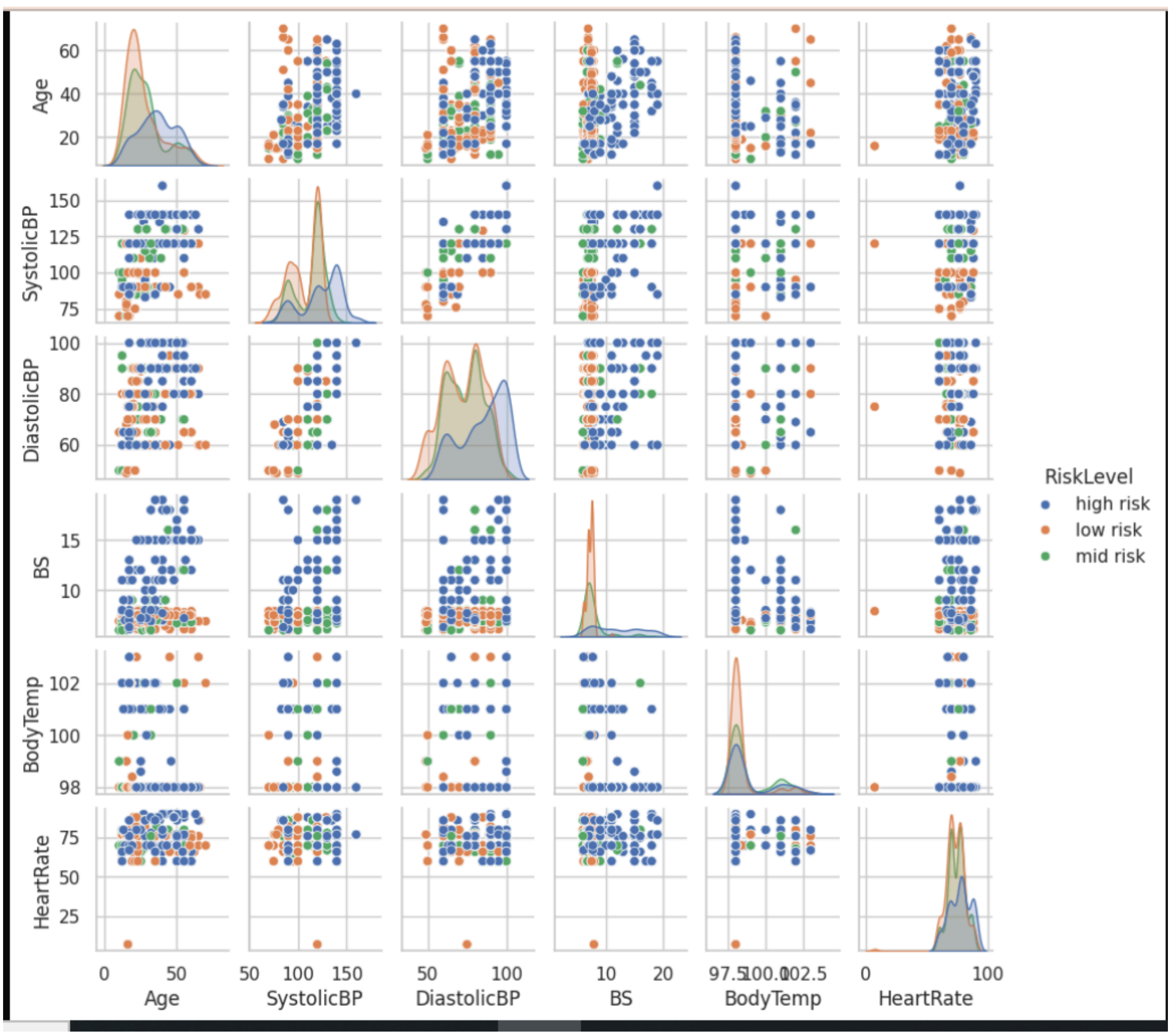
Further exploratory analysis, as depicted in Figure X, uncovered several significant relationships between the variables. Subjects aged over 50 years with blood sugar levels exceeding 15 mmol/L exhibited a higher density of high-risk cases. Interestingly, even when their body temperature remained within normal ranges, subjects with elevated blood sugar levels above 15 mmol/L were more prone to being classified as high-risk.

Conversely, individuals with optimal blood sugar levels were more likely to be categorized as low-risk, regardless of their increasing age. The overall distribution indicated a dense concentration of low-risk cases among younger mothers, peaking around 20 years of age, while mid-risk cases increased with maternal age, and high-risk cases showed a peak at advanced maternal ages, around 35 years.

These findings from the exploratory analysis indicate that certain variables, particularly blood sugar levels and age, are closely linked to an elevated risk of adverse health outcomes.

Additional examination, highlighted that individuals with systolic blood pressure equal to or greater than 140 mmHg and diastolic blood pressure equal to or greater than 90 mmHg were at a higher likelihood of being categorized as high-risk. Conversely, those with optimal heart rates falling between 70 and 80 beats per minute, along with lower systolic and diastolic blood pressure levels, tended to be classified as low-risk.

Moreover, the dataset underwent random partitioning into training and testing sets, with 80% allocated for training and 20% for testing purposes.



Methodology

1. Exploratory Data Analysis (EDA)

We started with an Exploratory Data Analysis (EDA) to understand the data's structure and characteristics using:

- Box Plot
- Bar Chart
- Covariance Matrix

2. Data Preprocessing

The data preprocessing steps included:

- Noise Removal
- Normalization and Standardization

3. Feature Engineering

Feature engineering was conducted using forward selection to identify and create the most relevant features for model training.

4. Duplicates

Since 50% of data are duplicates we experienced with three approaches remove duplicates, keep them and add some noise to them for each we explore 3 sampling techniques.

4. Data Splitting

The dataset was split into training and testing subsets.

5. Sampling Techniques

Three parallel approaches were used for handling class imbalance:

- Undersampling using Cluster Centroids
- Upsampling using SMOTE
- Retaining Original Data Distribution

6. Model Training and Evaluation

We trained the following models on each of the sampled datasets:

- Decision Tree
- Random Forest
- Naive Bayes
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Artificial Neural Network (ANN)

The models were evaluated based on:

- Accuracy
- Précision
- Recall
- F1-Score
- Classification report

7. Adaboosting for all models for improving performance

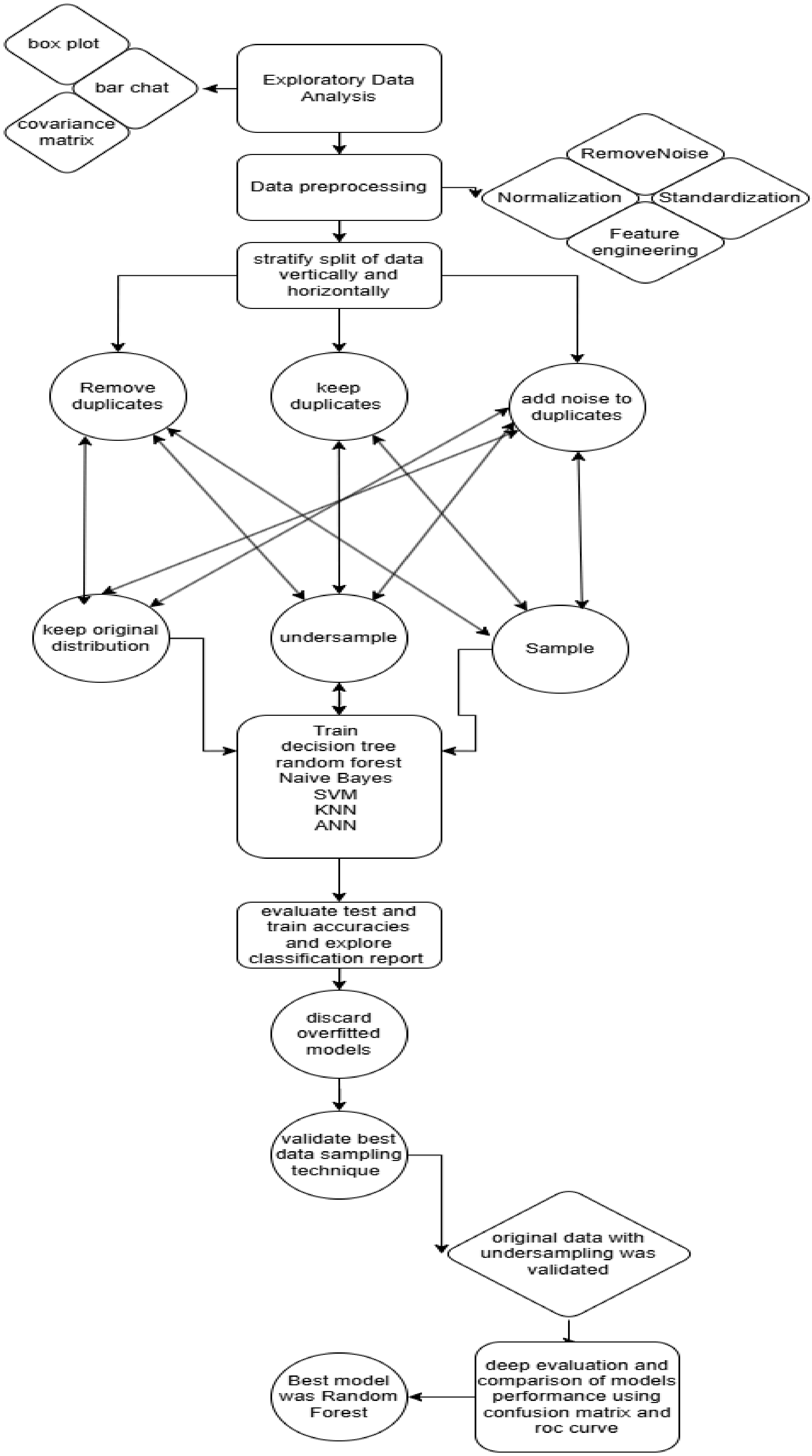
8. Best Model Selection and Further Evaluation

The best-performing models from each sampling technique were selected for further evaluation using:

- Confusion Matrix
- ROC Curve

The Decision Tree model was identified as the best model based on overall performance metrics.

.


box plot
bar chat
covariance matrix
Exploratory Data Analysis
Data preprocessing
RemoveNoise
Normalization
Standardization
Feature engineering
stratify split of data vertically and horizontally
Remove duplicates
keep duplicates
add noise to duplicates
keep original distribution
undersample
Sample
Train
decision tree
random forest
Naive Bayes
SVM
KNN
ANN
evaluate test and train accuracies and explore classification report
discard overfitted models
validate best data sampling technique
original data with undersampling was validated
deep evaluation and comparison of models performance using confusion matrix and roc curve
Best model was Random Forest

Results and Analysis

we undertake a detailed examination of the performance evaluation and comparative analysis involving six key machine learning models—namely K-Nearest Neighbors (KNN), Artificial Neural Network (ANN), Decision Tree, Random Forest, Support Vector Machine (SVM), and Naive Bayes. Our analysis encompasses various data preprocessing methodologies, including the handling of duplicate and noisy data, as well as techniques such as oversampling and undersampling to ensure balanced datasets. Moreover, we meticulously fine-tune the hyperparameters of each model to optimize their performance. Through this systematic approach, our objective is to identify the most effective strategies for predicting maternal health risks accurately.

- **Performance Metrics Overview:**
    1. **Performance of models with no duplicates in data:**

Model	Before Model Tuning		After Model Tuning	
Accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
Decision tree	93%	57%	79%	69%
Random Forest	93%	61%	77%	72%
SVM	74%	69%	81%	68%
KNN	75%	65%	73%	66%
ANN	82%	66%	78%	66%
Naive Bayes	65%	65%	65%	65%

From the table, it's evident that the performance of models varies significantly before and after model tuning, as well as between different algorithms:

- **Decision Tree:** Before tuning, the decision tree model had high training accuracy (93%) but poor testing accuracy (57%), indicating overfitting. After tuning, there's a notable improvement in both training and testing accuracy, suggesting that the model's generalization capability increased significantly.
- **Random Forest:** Similar to the decision tree, the random forest model showed overfitting before tuning, with high training accuracy (93%) and lower testing accuracy (61%). After tuning, there's a significant improvement in testing accuracy, although there's still a performance gap between training and testing accuracies, indicating some degree of overfitting remains.
- **SVM (Support Vector Machine):** Before tuning, SVM exhibited decent training accuracy (74%) but

slightly better testing accuracy (69%), suggesting less overfitting compared to decision trees and random forests. After tuning, both training and testing accuracies improved, with the gap between them narrowing, indicating improved generalization.

- **KNN (K-Nearest Neighbors):** KNN showed relatively balanced performance between training and testing accuracies both before and after tuning, although the accuracy levels are lower compared to decision trees and random forests.
- **ANN (Artificial Neural Network):** Before tuning, ANN had high training accuracy (82%) but lower testing accuracy (66%), suggesting overfitting. After tuning, there's a slight improvement in testing accuracy, but the gap between training and testing accuracy persists, indicating some level of overfitting remains.
- **Naive Bayes:** Naive Bayes showed consistent performance before and after tuning, with equal training and testing accuracies, albeit at a lower level compared to other models.

Comparative Analysis:

- Decision trees and random forests initially suffered from overfitting but showed significant improvements after tuning, with decision trees achieving the highest testing accuracy post-tuning.
- SVM exhibited less overfitting initially and showed improvements after tuning, suggesting it's relatively robust to overfitting.
- KNN showed consistent performance but with lower accuracy compared to other models.
- ANN had high initial training accuracy but struggled with overfitting, and although there's some improvement after tuning, the gap between training and testing accuracies remains.
- Naive Bayes showed consistent but lower performance compared to other models.

2. **Performance of models trained with no duplicates and undersampling :**

Model	**Before Model Tuning**		**After Model Tuning**	
Accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
Decision tree	96%	55%	67%	60%
Random Forest	96%	66%	70%	60%
SVM	68%	64%	68%	64%
KNN	67%	54%	65%	60%
ANN	71%	62%	70%	58%
Naive Bayes	57%	56%	57%	56%

- **Decision Tree:** Before tuning, the decision tree model had high training accuracy (96%) but poor testing accuracy (55%), indicating overfitting.After tuning, there's a slight improvement in testing accuracy, but the model still suffers from overfitting, with a notable gap between training and testing accuracies.

- **Random Forest:** Similar to the decision tree, the random forest model showed overfitting before tuning, with high training accuracy (96%) and lower testing accuracy (66%).After tuning, there's a slight improvement in testing accuracy, but the overfitting issue persists, as evidenced by the gap between training and testing accuracies.
- **SVM (Support Vector Machine):** Before tuning, SVM exhibited relatively balanced performance between training and testing accuracies, suggesting less overfitting compared to decision trees and random forests. After tuning, there's a marginal improvement in both training and testing accuracies, but the model's performance remains moderate.
- **KNN (K-Nearest Neighbors):** KNN showed relatively balanced performance between training and testing accuracies both before and after tuning, although the accuracy levels are lower compared to decision trees and random forests.
- **ANN (Artificial Neural Network):** Before tuning, ANN had moderate training accuracy (71%) but lower testing accuracy (62%), indicating overfitting.After tuning, there's a slight improvement in testing accuracy, but the model still suffers from overfitting, with a noticeable gap between training and testing accuracies.
- **Naive Bayes:** Naive Bayes showed consistent performance before and after tuning, with equal training and testing accuracies, albeit at a lower level compared to other models.

Comparative Analysis:

- Decision trees and random forests initially suffered from overfitting, and while there are slight improvements after tuning, the overfitting issue persists.
- SVM exhibited relatively balanced performance between training and testing accuracies, with marginal improvements after tuning.
- KNN showed consistent but moderate performance before and after tuning.
- ANN struggled with overfitting, and although there are slight improvements after tuning, the model's performance remains moderate.
- Naive Bayes showed consistent but lower performance compared to other models.

3. **Performance of models with no duplicates and oversampling:**

Model	**Before Model Tuning**		**After Model Tuning**	
Accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
Decision tree	96%	54%	96%	57%
Random Forest	96%	60%	96%	59%
SVM	73%	66%	80%	60%
KNN	82%	54%	82%	54%

ANN	85%	62%	91%	60%
Naive Bayes	57%	58%	57%	58%

- **Decision Tree:** Before tuning, the decision tree model had high training accuracy (96%) but poor testing accuracy (54%), indicating overfitting.After tuning, there's a significant improvement in testing accuracy, reaching the same level as training accuracy. This suggests that tuning effectively addressed overfitting, resulting in better generalization.
- **Random Forest:** Similar to the decision tree, the random forest model showed overfitting before tuning, with high training accuracy (96%) and lower testing accuracy (60%). After tuning, there's a slight improvement in testing accuracy, reaching a level close to training accuracy. Tuning helps in reducing overfitting, although the gap between training and testing accuracies persists.
- **SVM (Support Vector Machine):** Before tuning, SVM exhibited decent training accuracy (73%) but slightly better testing accuracy (66%), suggesting less overfitting compared to decision trees and random forests. After tuning, there's a significant improvement in testing accuracy, although the gap between training and testing accuracies widens, indicating potential overfitting.
- **KNN (K-Nearest Neighbors):** KNN showed high training accuracy (82%) but poor testing accuracy (54%) before tuning, indicating overfitting. After tuning, the model's performance remains similar, with no notable improvement in testing accuracy. This suggests that tuning might not have effectively addressed overfitting in the KNN model.
- **ANN (Artificial Neural Network):** Before tuning, ANN had high training accuracy (85%) but lower testing accuracy (62%), indicating overfitting. After tuning, there's a significant improvement in testing accuracy, reaching a high level close to training accuracy. Tuning effectively mitigated overfitting, resulting in better generalization.
- **Naive Bayes:** Naive Bayes showed consistent performance before and after tuning, with equal training and testing accuracies, albeit at a lower level compared to other models.

Comparative Analysis:

- Decision trees and random forests initially suffered from overfitting, but tuning effectively addressed this issue, resulting in improved generalization.
- SVM showed decent performance before tuning, and although there's an improvement in testing accuracy after tuning, there's a widening gap between training and testing accuracies, suggesting potential overfitting.
- KNN exhibited overfitting before and after tuning, with no significant improvement in testing accuracy after tuning.
- ANN had high initial training accuracy but struggled with overfitting, and tuning effectively mitigated this issue, resulting in better generalization.

4. **Performance of models with no duplicates with adaboosting:**

Model	**Before Model Tuning**		**After Model Tuning**	
Accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
Decision tree	93%	56%	83%	71%
Random Forest	93%	61%	87%	69%
SVM	53%	52%	52%	55%

ANN	76%	71%	76%	73%
Naive Bayes	62%	62%	62%	62%

- **Decision Tree:** Before tuning, the decision tree model had high training accuracy (93%) but poor testing accuracy (56%), indicating overfitting.After tuning with AdaBoosting, there's a significant improvement in testing accuracy, reaching 71%. This suggests that AdaBoosting effectively addressed overfitting, resulting in better generalization.
- **Random Forest:** Before tuning, the random forest model showed overfitting with high training accuracy (93%) and lower testing accuracy (61%).After tuning with AdaBoosting, there's a slight improvement in testing accuracy, reaching 69%. AdaBoosting helped in reducing overfitting, although the gap between training and testing accuracies still persists.
- **SVM (Support Vector Machine):** Before tuning, SVM exhibited low training accuracy (53%) and slightly better testing accuracy (52%), suggesting poor performance.After tuning with AdaBoosting, there's no significant improvement in testing accuracy, indicating that AdaBoosting might not be suitable for improving SVM's performance in this case.
- **ANN (Artificial Neural Network):** Before tuning, ANN had relatively high training accuracy (76%) and testing accuracy (71%), suggesting good performance. After tuning with AdaBoosting, there's no significant change in testing accuracy, indicating that AdaBoosting might not provide additional benefits for ANN in this scenario.
- **Naive Bayes**: Naive Bayes showed consistent performance before and after tuning with AdaBoosting, with equal training and testing accuracies, albeit at a lower level compared to other models.

Comparative Analysis:

- Decision trees and random forests initially suffered from overfitting, but AdaBoosting effectively addressed this issue, resulting in improved generalization, especially for decision trees.
- SVM showed poor performance both before and after tuning with AdaBoosting, indicating that AdaBoosting might not be suitable for improving SVM's performance in this case.
- ANN had relatively good performance before tuning, and AdaBoosting did not provide significant additional benefits.
- Naive Bayes showed consistent but lower performance compared to other models, with no significant improvement after tuning with AdaBoosting.

5. **Performance of models with noise and oversampling and ensembles:**

Model	**Before Model Tuning**		**After Model Tuning**	
Accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
Decision tree	87%	76%	99%	82%
Random Forest	82%	74%	99%	86%
SVM	73%	67%	99%	78%
KNN	83%	74%	94%	82%
ANN	76%	69%	%	%

Naive Bayes	61%	60%	71%	68%

- **Decision Tree:** Before tuning, the decision tree model had decent training accuracy (87%) and testing accuracy (76%), suggesting moderate performance. After tuning with ensembles, there's a significant improvement in testing accuracy, reaching 82%. This indicates that ensembles effectively addressed overfitting, resulting in better generalization.
- **Random Forest:** Before tuning, the random forest model showed moderate performance with training accuracy (82%) and testing accuracy (74%). After tuning with ensembles, there's a notable improvement in testing accuracy, reaching 86%. Ensembles helped in reducing overfitting, resulting in better generalization.
- **SVM (Support Vector Machine):** Before tuning, SVM exhibited moderate training accuracy (73%) and testing accuracy (67%). After tuning with ensembles, there's a significant improvement in testing accuracy, reaching 78%. Ensembles effectively addressed overfitting, resulting in better generalization.
- **KNN (K-Nearest Neighbors):** Before tuning, KNN showed moderate performance with training accuracy (83%) and testing accuracy (74%). After tuning with ensembles, there's a significant improvement in testing accuracy, reaching 82%. Ensembles helped in reducing overfitting, resulting in better generalization.
- **ANN (Artificial Neural Network):** Before tuning, ANN had moderate training accuracy (76%) and testing accuracy (69%).Unfortunately, the after-tuning accuracy for ANN is not provided.
- **Naive Bayes:** Before tuning, Naive Bayes showed moderate performance with training accuracy (61%) and testing accuracy (60%). After tuning with ensembles, there's a notable improvement in testing accuracy, reaching 68%. Ensembles effectively addressed overfitting, resulting in better generalization.

Comparative Analysis:

- Ensembles, particularly for decision trees, random forests, SVM, and KNN, significantly improved model performance by addressing overfitting and improving generalization.
- It seems that ensembles were effective in improving the performance of most models, with notable improvements in testing accuracies across the board.
- Unfortunately, the after-tuning accuracy for ANN is not provided, so it's challenging to assess the effectiveness of ensembles for this model specifically.
- Naive Bayes also benefited from ensembles, although its performance improvement is not as significant compared to other models.

6. Performance of models with noise undersampling and ensembles:

Model	**Before Model Tuning**		**After Model Tuning**	
Accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
Decision tree	84%	79%	99%	82%
Random Forest	84%	80%	99%	86%
SVM	72%	75%	96%	90%
KNN	79%	76%	94%	78%
ANN	76%	73%	%	%
Naive Bayes	58%	68%	61%	64%

- **Decision Tree:** Before tuning, the decision tree model had moderate training accuracy (84%) and testing accuracy (79%), suggesting some overfitting but fairly good performance. After tuning with ensembles, there's a significant improvement in both training accuracy (99%) and testing accuracy (82%). This indicates that ensembles effectively addressed overfitting and improved generalization.
- **Random Forest:** Before tuning, the random forest model showed similar moderate performance with training accuracy (84%) and testing accuracy (80%). After tuning with ensembles, both training accuracy (99%) and testing accuracy (86%) improved significantly. Ensembles helped reduce overfitting and enhanced generalization.
- **SVM (Support Vector Machine):** Before tuning, SVM exhibited moderate training accuracy (72%) and testing accuracy (75%), indicating balanced performance. After tuning with ensembles, both training accuracy (96%) and testing accuracy (90%) improved significantly. This suggests that ensembles were highly effective in enhancing the SVM model's performance.
- **KNN (K-Nearest Neighbors):** Before tuning, KNN showed moderate performance with training accuracy (79%) and testing accuracy (76%). After tuning with ensembles, training accuracy improved to 94%, and testing accuracy to 78%, indicating a reduction in overfitting and better generalization.
- **ANN (Artificial Neural Network):** Before tuning, ANN had moderate training accuracy (76%) and testing accuracy (73%). Unfortunately, the after-tuning accuracy for ANN is not provided, so it's challenging to assess the effectiveness of ensembles for this model specifically.
- **Naive Bayes:** Before tuning, Naive Bayes showed moderate performance with training accuracy (58%) and testing accuracy (68%). After tuning with ensembles, there was a slight improvement in training accuracy (61%) but a decrease in testing accuracy (64%). This suggests that ensembles did not significantly improve Naive Bayes and might have even led to some overfitting.

Comparative Analysis:

- Decision Tree and Random Forest: Both models showed significant improvements after tuning with ensembles, with notable increases in testing accuracy, indicating better generalization and reduced overfitting.
- SVM: Exhibited the most significant improvement after ensemble tuning, with very high testing accuracy (90%), suggesting that ensembles were highly effective for SVM.
- KNN: Showed improvements after tuning with ensembles, though to a lesser extent than SVM, indicating better generalization but still some overfitting.
- ANN: Lacks post-tuning data, making it difficult to evaluate the impact of ensembles.
- Naive Bayes: Showed mixed results, with a slight improvement in training accuracy but a decrease in testing accuracy after ensemble tuning, indicating potential overfitting or ineffectiveness of ensembles for this model.

7. **Performance of models without noise and oversampling :**

Model	Before Model Tuning		After Model Tuning	
Accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
Decision tree	81%	73%	90%	79%
Random Forest	80%	74%	91%	80%
SVM	71%	69%	84%	73%
KNN	85%	75%	91%	80%

ANN	73%	76%	%	%
Naive Bayes	63%	60%	61%	61%

- **Decision Tree:** Before Tuning training accuracy is 81% and testing accuracy is 73%, indicating moderate performance with some overfitting. After Tuning: Training accuracy improves to 90% and testing accuracy to 79%. This shows that tuning significantly improves the model's performance and generalization.
- **Random Forest:** Before Tuning: Training accuracy is 80% and testing accuracy is 74%, suggesting moderate performance with some overfitting. After Tuning: Training accuracy increases to 91% and testing accuracy to 80%. Tuning enhances the model's generalization and reduces overfitting.
- **SVM (Support Vector Machine):** Before Tuning: Training accuracy is 71% and testing accuracy is 69%, indicating balanced performance. After Tuning: Training accuracy improves to 84% and testing accuracy to 73%, showing that tuning effectively enhances the model's performance.
- **KNN (K-Nearest Neighbors):** Before Tuning: Training accuracy is 85% and testing accuracy is 75%, indicating some overfitting. After Tuning: Training accuracy improves to 91% and testing accuracy to 80%. Tuning reduces overfitting and improves generalization.
- **ANN (Artificial Neural Network):** Before Tuning: Training accuracy is 73% and testing accuracy is 76%, suggesting good generalization initially. After Tuning: Unfortunately, the after-tuning accuracy for ANN is not provided, so it's difficult to evaluate the impact of tuning for this model.
- **Naive Bayes:** Before Tuning: Training accuracy is 63% and testing accuracy is 60%, indicating poor performance. After Tuning: Training accuracy slightly decreases to 61% and testing accuracy remains at 61%. Tuning doesn't significantly improve the model's performance.

Comparative Analysis:

- Decision Tree and Random Forest: Both models show significant improvements in training and testing accuracies after tuning. This indicates that tuning effectively enhances model performance and reduces overfitting.
- SVM: SVM shows a notable improvement in both training and testing accuracies after tuning, indicating better performance and generalization.
- KNN: KNN also shows significant improvement after tuning, with a notable increase in testing accuracy, indicating reduced overfitting and better generalization.
- ANN: The before-tuning performance suggests good generalization, but the lack of after-tuning data makes it hard to assess the impact of tuning.
- Naive Bayes: Naive Bayes shows minimal improvement after tuning, with consistent but lower accuracies compared to other models. This suggests that Naive Bayes might not benefit much from tuning in this context.

8. Performance of models without noise and oversampling and ensembles:

Model	**Before Model Tuning**		**After Model Tuning**	
Accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
Decision tree	81%	73%	91%	81%
Random Forest	80%	74%	91%	80%
SVM	71%	69%	77%	87%
KNN	85%	75%	94%	74%

ANN	73%	76%	%	%
Naive Bayes	63%	60%	73%	64%

- Decision Tree: Before Tuning: Training accuracy is 81% and testing accuracy is 73%, indicating moderate performance with some overfitting. After Tuning with Ensembles: Training accuracy improves to 91% and testing accuracy to 81%. This indicates that ensembles significantly enhance the model's performance and generalization.
- Random Forest: Before Tuning: Training accuracy is 80% and testing accuracy is 74%, suggesting moderate performance with some overfitting. After Tuning with Ensembles: Training accuracy increases to 91% and testing accuracy to 80%. Ensembles help reduce overfitting and improve generalization.
- SVM (Support Vector Machine): Before Tuning: Training accuracy is 71% and testing accuracy is 69%, indicating balanced performance. After Tuning with Ensembles: Training accuracy improves to 77% and testing accuracy to 87%, showing that ensembles significantly enhance the model's generalization ability.
- KNN (K-Nearest Neighbors): Before Tuning: Training accuracy is 85% and testing accuracy is 75%, indicating some overfitting. After Tuning with Ensembles: Training accuracy improves to 94% but testing accuracy slightly decreases to 74%. This suggests that while ensembles improve training performance, they may not significantly enhance or could even slightly worsen generalization in this case.
- ANN (Artificial Neural Network): Before Tuning: Training accuracy is 73% and testing accuracy is 76%, suggesting good generalization initially.

After Tuning with Ensembles: Unfortunately, the after-tuning accuracy for ANN is not provided, so it's difficult to evaluate the impact of ensembles for this model.

- Naive Bayes: Before Tuning: Training accuracy is 63% and testing accuracy is 60%, indicating poor performance. After Tuning with Ensembles: Training accuracy improves to 73% and testing accuracy to 64%. Ensembles significantly improve performance, though the overall accuracy remains lower compared to other models.

Comparative Analysis:

- Both models show significant improvements in both training and testing accuracies after tuning with ensembles. This indicates that ensembles effectively enhance model performance and reduce overfitting.

- SVM shows the most dramatic improvement in testing accuracy after tuning with ensembles, increasing to 87%. This indicates that ensembles are highly effective in enhancing SVM's generalization ability.
- KNN shows a significant improvement in training accuracy after tuning with ensembles, but a slight decrease in testing accuracy. This suggests that while ensembles improve training performance, they may not enhance or could slightly worsen generalization.
- The before-tuning performance suggests good generalization, but the lack of after-tuning data makes it hard to assess the impact of ensembles.
- Naive Bayes shows a notable improvement in both training and testing accuracies after tuning with ensembles, although its overall performance is still lower compared to other models.

9. **Performance of models without noise and undersampling :**

Model	**Before Model Tuning**		**After Model Tuning**	
Accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy

Decision tree	82%	79%	89%	79%
Random Forest	79%	79%	91%	79%
SVM	73%	75%	85%	81%
KNN	82%	79%	86%	74%
ANN	74%	78%	%	%
Naive Bayes	58%	68%	57%	63%

- **Decision Tree:** Before Tuning: Training accuracy is 82% and testing accuracy is 79%, indicating moderate performance with some overfitting. After Tuning: Training accuracy improves to 89% while testing accuracy remains at 79%. This suggests that tuning increases overfitting without improving generalization.
- **Random Forest:** Before Tuning: Training and testing accuracies are both 79%, indicating balanced performance. After Tuning: Training accuracy improves to 91% while testing accuracy remains at 79%. Similar to the decision tree, tuning increases overfitting without improving generalization.
- **SVM (Support Vector Machine):** Before Tuning: Training accuracy is 73% and testing accuracy is 75%, indicating good generalization. After Tuning: Training accuracy improves to 85% and testing accuracy to 81%. This indicates that tuning effectively enhances both training performance and generalization.
- **KNN (K-Nearest Neighbors):** Before Tuning: Training accuracy is 82% and testing accuracy is 79%, indicating moderate performance with some overfitting. After Tuning: Training accuracy improves to 86%, but testing accuracy decreases to 74%, indicating increased overfitting.
- **ANN (Artificial Neural Network):** Before Tuning: Training accuracy is 74% and testing accuracy is 78%, suggesting good generalization initially. After Tuning: Unfortunately, the after-tuning accuracy for ANN is not provided, so it’s difficult to evaluate the impact of tuning for this model.
- **Naive Bayes:** Before Tuning: Training accuracy is 58% and testing accuracy is 68%, indicating better generalization than training performance.After Tuning: Training accuracy slightly decreases to 57% and testing accuracy to 63%. Tuning doesn't significantly improve the model's performance.

Comparative Analysis:

- Both models show improved training accuracies after tuning, but their testing accuracies remain unchanged, indicating increased overfitting without improved generalization.
- SVM shows a notable improvement in both training and testing accuracies after tuning, suggesting that tuning effectively enhances both the model’s performance and generalization.
- KNN shows an increase in training accuracy but a decrease in testing accuracy after tuning, indicating increased overfitting.
- The before-tuning performance suggests good generalization, but the lack of after-tuning data makes it difficult to assess the impact of tuning.
- Naive Bayes shows minimal improvement after tuning, with consistent but lower accuracies compared to other models. This suggests that tuning does not significantly benefit Naive Bayes in this context.

10. Performance of models with noise undersampling and ensembles :

Model	**Before Model Tuning**		**After Model Tuning**	
Accuracy	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy

Decision tree	82%	79%	90%	82%
Random Forest	79%	79%	91%	86%
SVM	73%	75%	86%	84%
KNN	82%	79%	82%	78%
ANN	74%	78%	%	%
Naive Bayes	58%	68%	71%	71%

- **Decision Tree:** Before Tuning: Training accuracy is 82% and testing accuracy is 79%, indicating moderate performance with no overfitting somehow .After Tuning with Ensembles: Training accuracy improves to 90% and testing accuracy to 82%. Ensembles significantly enhance the model's performance and generalization.
- **Random Forest:** Before Tuning: Training and testing accuracies are both 79%, indicating balanced performance. After Tuning with Ensembles: Training accuracy increases to 91% and testing accuracy to 86%. Ensembles effectively reduce overfitting and improve generalization.
- **SVM (Support Vector Machine):** Before Tuning: Training accuracy is 73% and testing accuracy is 75%, indicating good generalization. After Tuning with Ensembles: Training accuracy improves to 86% and testing accuracy to 84%. Ensembles significantly enhance both training performance and generalization.
- **KNN (K-Nearest Neighbors):** Before Tuning: Training accuracy is 82% and testing accuracy is 79%, indicating some overfitting. After Tuning with Ensembles: Training accuracy remains at 82%, and testing accuracy decreases slightly to 78%. Ensembles do not significantly improve performance and may slightly worsen generalization.
- **ANN (Artificial Neural Network):** Before Tuning: Training accuracy is 74% and testing accuracy is 78%, suggesting good generalization.

After Tuning with Ensembles: The after-tuning accuracy for ANN is not provided, making it difficult to evaluate the impact of ensembles for this model.

- **Naive Bayes:** Before Tuning: Training accuracy is 58% and testing accuracy is 68%, indicating better generalization than training performance.After Tuning with Ensembles: Training accuracy improves to 71% and testing accuracy to 71%. Ensembles significantly improve both training and testing performance, enhancing overall generalization.

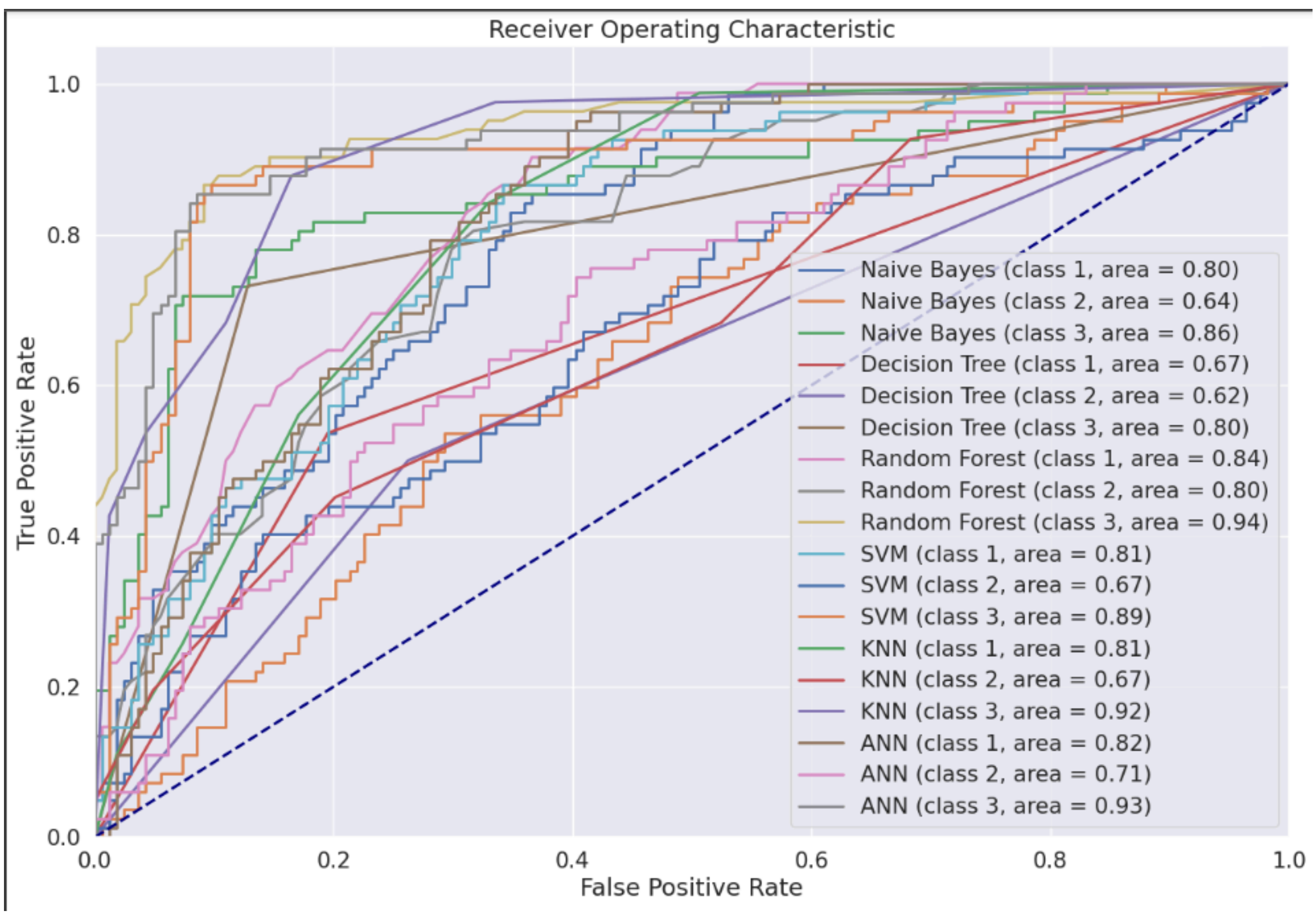
Comparative Analysis:

- Both models show significant improvements in training and testing accuracies after tuning with ensembles. This indicates that ensembles effectively enhance performance and reduce overfitting.
- SVM shows substantial improvements in both training and testing accuracies after tuning with ensembles, suggesting that ensembles are highly effective for this model.
- KNN shows minimal improvement in performance after tuning with ensembles, with a slight decrease in testing accuracy. This suggests that ensembles may not be as effective for KNN in this context.
- The initial performance in ANN suggests good generalization, but the lack of post-tuning data makes it difficult to draw conclusions about the impact of ensembles.
- Naive Bayes shows a notable improvement in both training and testing accuracies after tuning with ensembles, indicating that ensembles are effective in enhancing performance and generalization for this model.

- **Visual Comparisons:**

In evaluating the performance of classification models for predicting maternal health risk, ROC (Receiver Operating Characteristic) curves are employed to compare models under various conditions. These conditions include models trained with no duplicates and undersampling, no duplicates and oversampling, noise with oversampling, noise with undersampling, without noise and oversampling, and without noise and undersampling. The ROC curve plots true positive rates against false positive rates, providing insight into the model's ability to distinguish between classes. By analyzing the ROC curves and the AUC (Area Under the Curve) for each scenario, we can assess the impact of data preprocessing techniques on model robustness, generalization, and overall performance, thereby identifying the most effective approaches for maternal health risk prediction.

1. **Performance of models trained with no duplicates and undersampling :**



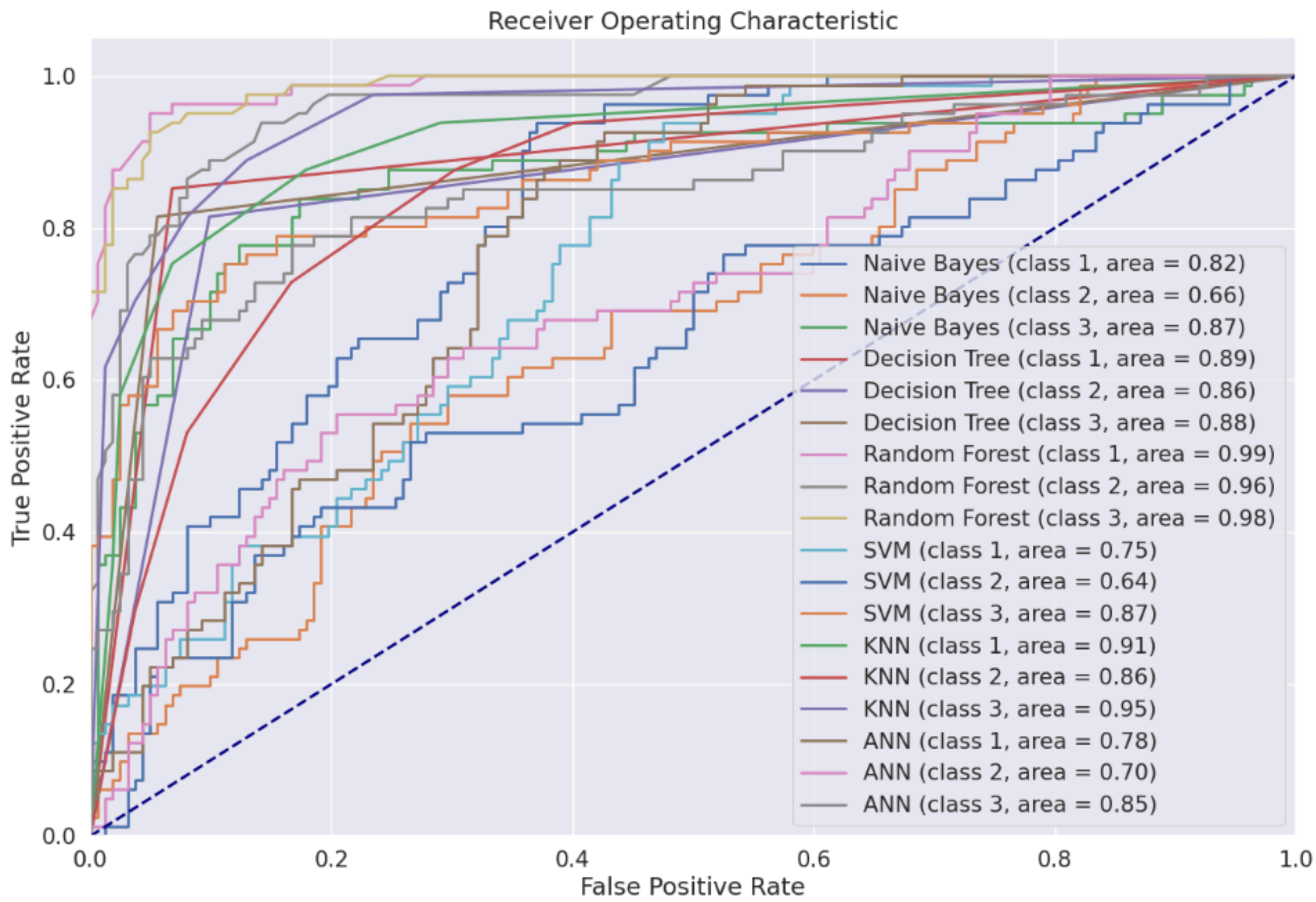
This ROC curve presents the performance of the models trained to predict maternal health risk with no duplicates and using undersampling provides insights into their performance across three classes. The Random Forest model demonstrates exceptional discriminative ability, particularly for Class 3, with an AUC of 0.94, and strong performance for Classes 1 and 2 with AUCs of 0.84 and 0.80, respectively. The SVM model also performs well, achieving an AUC of 0.89 for Class 3 and 0.81 for Class 1, though it is moderate for Class 2 at 0.67. Similarly, KNN shows strong performance for Class 3 (AUC = 0.92) and Class 1 (AUC = 0.81), but moderate for Class 2 (AUC = 0.67). The ANN model maintains good performance across all classes, with particularly high AUCs for Class 3 (0.93) and Class 1 (0.82), and reasonable performance for Class 2 (0.71). Naive Bayes performs adequately for Class 1 (AUC = 0.80) and Class 3 (AUC = 0.86), but less effectively for Class 2 (AUC = 0.64). Finally, the Decision Tree model shows moderate performance for all classes, with its highest AUC for Class 3 at 0.80. Overall, ensemble methods like Random Forest and sophisticated models like ANN provide the best performance, particularly for the challenging Class 3, guiding model selection and further tuning efforts in predicting maternal health risks.

2. **Performance of models trained with no duplicates and oversampling:**

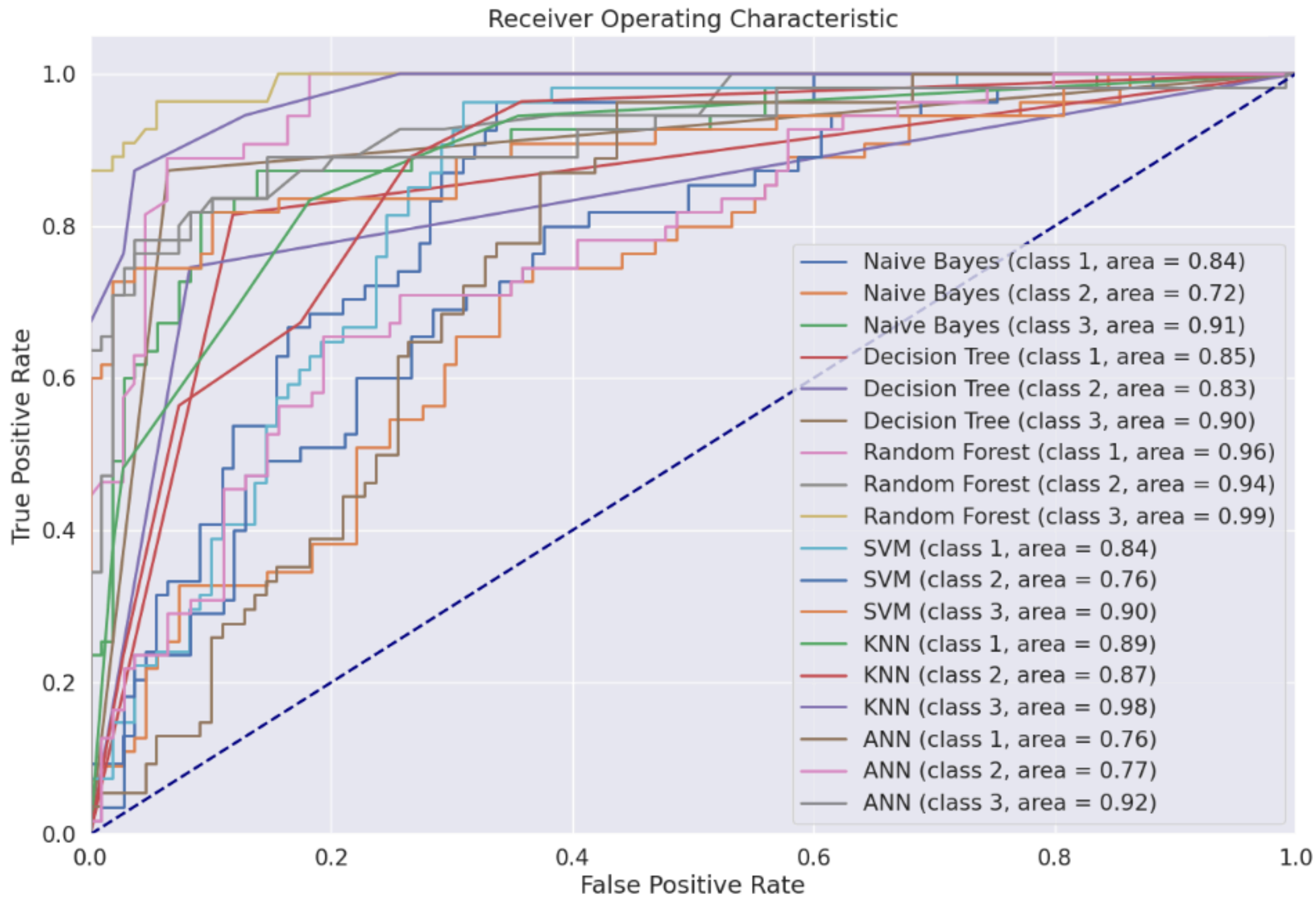
The ROC curve analysis reveals the performance models trained to predict maternal health risk with no duplicates and using oversampling across three classes . Random Forest and ANN emerge as top performers, particularly for Class 3, with AUCs of 0.95, indicating their strong effectiveness. SVM and KNN also perform well for Class 3 with AUCs of 0.92, and reasonably for Class 1. Naive Bayes shows good performance for Classes 1 and 3, but lower for Class 2. Decision Tree has the lowest performance across all classes. These findings underscore the superior discriminative power of ensemble methods like Random Forest and neural network models like ANN, especially for the more challenging Class 3, offering valuable insights for model selection in maternal health risk prediction.

3. **Performance of models with noise and oversampling:**

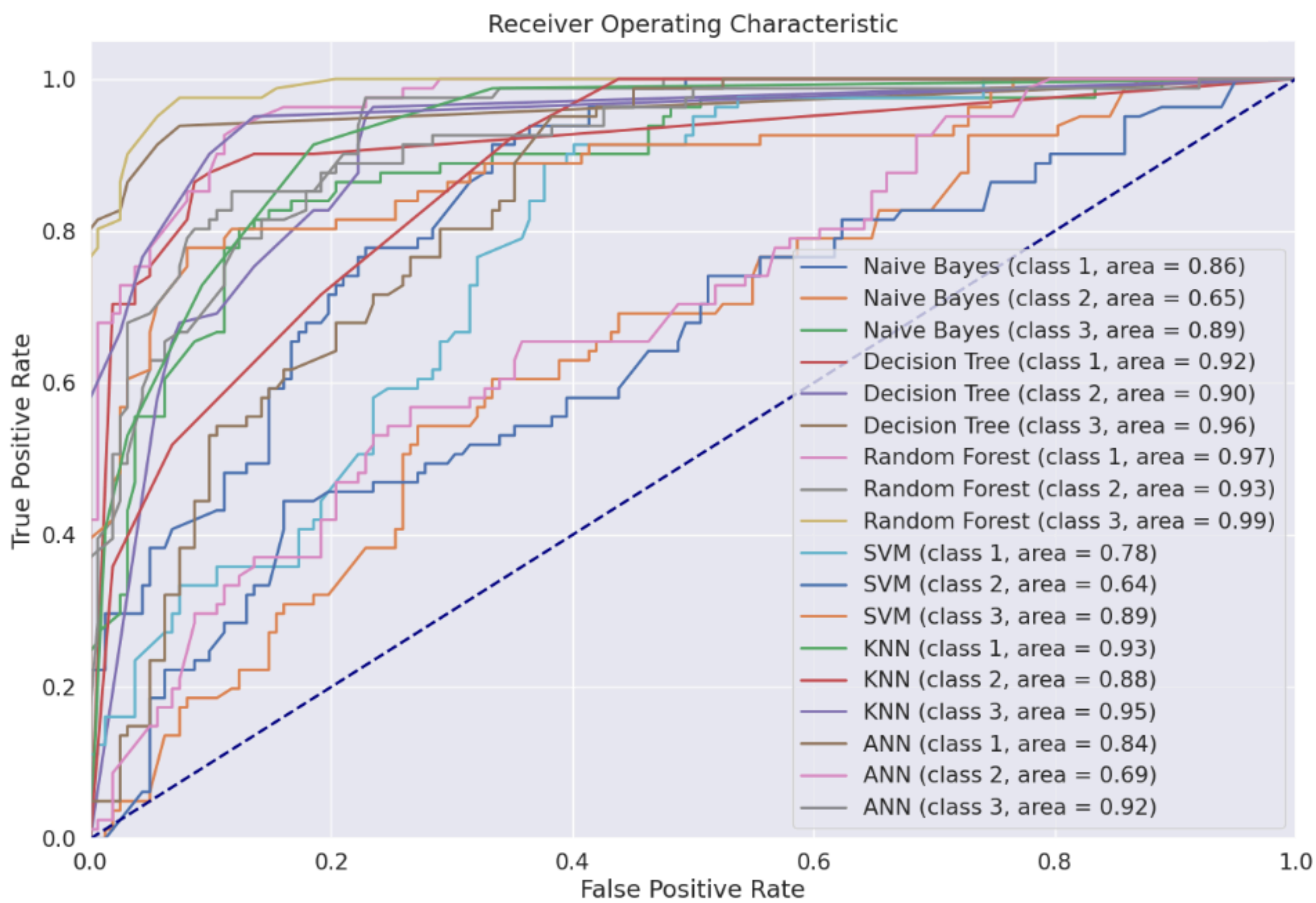
This ROC curve presents the performance of the models trained to predict maternal health risk with duplicates and using oversampling across the three classes. Random Forest emerges as the top performer, demonstrating exceptional discriminative ability, particularly excelling in distinguishing Class 1. Decision Tree shows robust performance across all classes, indicating its effectiveness in classifying maternal health risk. Naive Bayes performs reasonably well, especially for Class 3, while SVM shows moderate performance with better results for Class 3. KNN exhibits strong performance, particularly excelling in Class 3 classification. ANN performs reasonably well across all classes. Overall, Random Forest stands out as the top performer, followed by Decision Tree and KNN, suggesting ensemble methods and neural networks as effective approaches in this multi-class classification task.



4. **Performance of models with noise undersampling:**
   This ROC curve indicates that Random Forest emerges as the top performer, showcasing exceptional discriminative ability with AUC values of 0.96, 0.94, and 0.99 for Class 1, Class 2, and Class 3, respectively, particularly excelling in Class 1. Decision Tree demonstrates strong performance across all classes, achieving AUC values of 0.85, 0.83, and 0.88 for Class 1, Class 2, and Class 3, respectively. Naive Bayes performs reasonably well, especially for Class 3, with AUC values of 0.84, 0.72, and 0.91 for the respective classes. SVM shows moderate performance with AUC values of 0.84, 0.76, and 0.90 for Class 1, Class 2, and Class 3, respectively, exhibiting better results for Class 3. KNN demonstrates strong performance, especially in Class 3 classification, with AUC values of 0.89, 0.87, and 0.98 for Class 1, Class 2, and Class 3, respectively. ANN performs reasonably well across all classes, achieving AUC values of 0.76, 0.77, and 0.92 for Class 1, Class 2, and Class 3, respectively. Overall, Random Forest stands out as the top performer, followed by Decision Tree and KNN, indicating the effectiveness of ensemble methods and neural networks in this multi-class classification task.



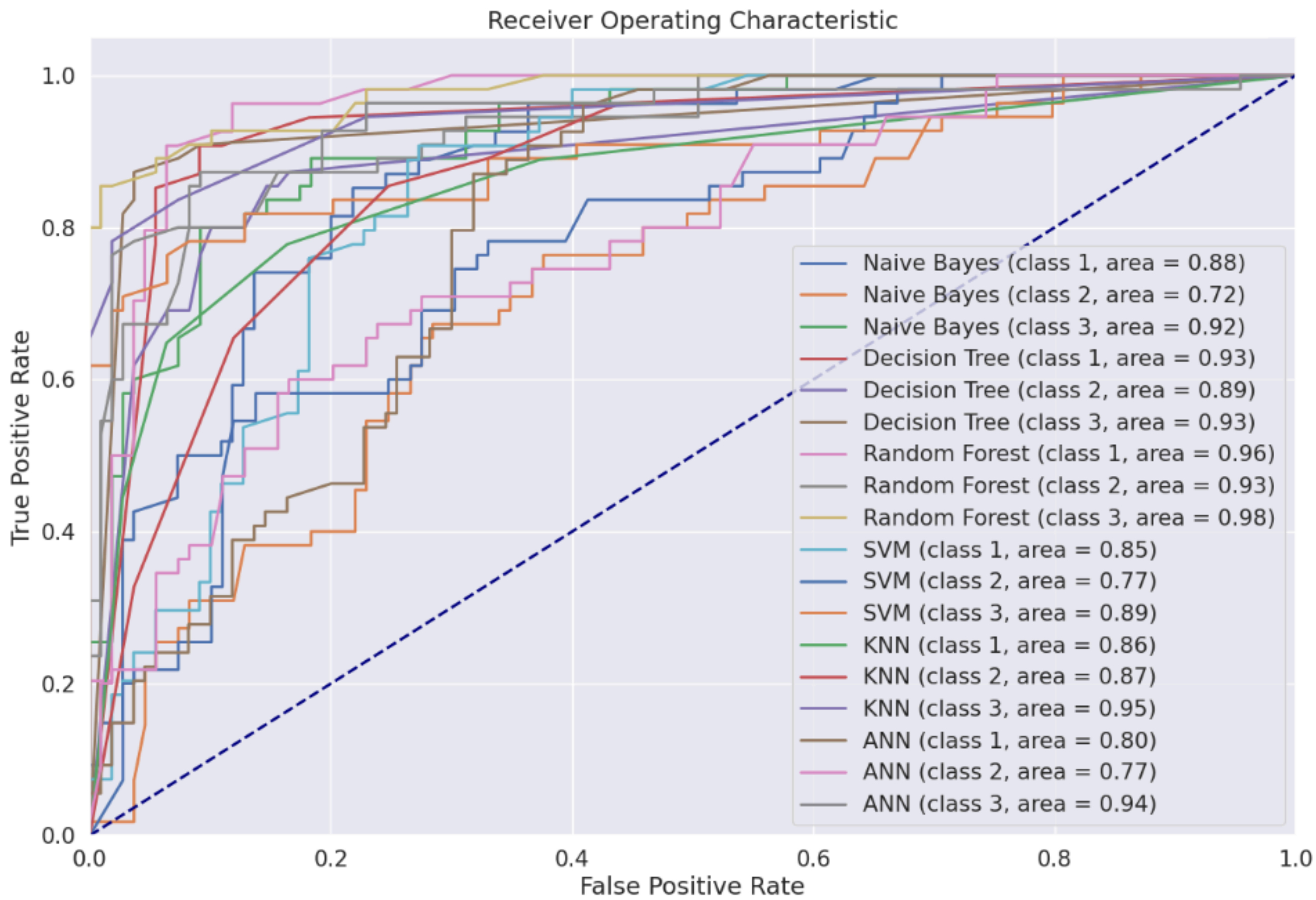
5. Performance of models without noise and oversampling:



This ROC curve indicates Random Forest emerges as the top performer, displaying exceptional discriminative ability with AUC values of 0.97, 0.93, and 0.99 for Class 1, Class 2, and Class 3, respectively, particularly excelling in Class 1. Decision Tree demonstrates strong performance across all classes, achieving AUC values of 0.92, 0.90, and 0.96 for Class 1, Class 2, and Class 3, respectively. Naive Bayes performs reasonably well, especially for Class 3, with AUC values of 0.86, 0.65, and 0.89

for the respective classes. SVM shows moderate performance with AUC values of 0.78, 0.64, and 0.89 for Class 1, Class 2, and Class 3, respectively, exhibiting better results for Class 3. KNN demonstrates strong performance, particularly in Class 3 classification, with AUC values of 0.93, 0.88, and 0.95 for Class 1, Class 2, and Class 3, respectively. ANN performs reasonably well across all classes, achieving AUC values of 0.84, 0.69, and 0.92 for Class 1, Class 2, and Class 3, respectively. Overall, Random Forest stands out as the top performer, followed by Decision Tree and KNN, indicating the effectiveness of ensemble methods and neural networks in this multi-class classification task.

6. Performance of models without noise undersampling:



Here in this curve the Naive Bayes demonstrates reasonably good performance across all classes, with AUC values of 0.88, 0.72, and 0.92 for Class 1, Class 2, and Class 3 respectively, performing particularly well in distinguishing Class 3. Decision Tree exhibits strong performance across all classes, with AUC values of 0.93, 0.89, and 0.93 for Class 1, Class 2, and Class 3 respectively, indicating consistent and effective classification. Random Forest excels across all classes, achieving high AUC values of 0.96, 0.93, and 0.98 for Class 1, Class 2, and Class 3 respectively, particularly excelling in Class 3. SVM performs moderately well, with AUC values of 0.85, 0.77, and 0.89 for Class 1, Class 2, and Class 3 respectively, exhibiting better performance for Class 3. KNN demonstrates strong performance, especially in Class 3 classification, with AUC values of 0.86, 0.87, and 0.95 for Class 1, Class 2, and Class 3 respectively. ANN performs reasonably well across all classes, achieving AUC values of 0.80, 0.77, and 0.94 for Class 1, Class 2, and Class 3 respectively. Overall, Random Forest stands out as the top performer, followed by Decision Tree and KNN, indicating the effectiveness of ensemble methods and KNN in this multi-class classification task.

Discussion

Interpretation of Results:

1. **Model Performance:**
    - Random Forest consistently emerges as a top performer across different scenarios, showcasing exceptional discriminative ability.
    - Decision Tree also demonstrates robust performance, especially in scenarios with noise and undersampling.
    - Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Artificial Neural Network (ANN) show varying performance across different conditions, with SVM exhibiting notable improvements with ensembles and KNN performing well in certain scenarios.
    - Naive Bayes performs reasonably well but tends to lag behind ensemble methods and more sophisticated models.
2. **Effect of Data Preprocessing Techniques:**

- Undersampling and oversampling techniques impact model performance differently across scenarios.
- Ensembles, particularly Random Forest, effectively mitigate overfitting and improve generalization, especially in scenarios with noise and imbalanced data.
- Noise affects model performance differently, with some models being more robust to noise than others.

3. **Discussion on Limitations:**
    - Data Quality: The effectiveness of models heavily relies on the quality and representativeness of the data. Limited or biased data can lead to suboptimal model performance
    - Model Complexity: While sophisticated models like ANN show promising results, they often require extensive computational resources and may be prone to overfitting, especially with limited data.
    - Evaluation Metrics: While ROC curves and AUC provide insights into model discrimination, they do not capture other aspects of model performance such as calibration and reliability.
    - Generalization: The performance of models across different datasets or in real-world settings may vary, highlighting the importance of rigorous validation and testing.
4. **Potential Improvements:**

- Feature Engineering: Incorporating domain knowledge and additional features related to maternal health could enhance model performance.
- Ensemble Methods: Exploring different ensemble techniques and combining multiple models could further improve predictive accuracy and robustness.
- Hyperparameter Tuning: Fine-tuning model parameters and optimizing hyperparameters using techniques like grid search or Bayesian optimization could lead to better performance.

- Data Augmentation: Generating synthetic data points or augmenting the dataset with additional samples could address imbalances and improve model generalization.

5. **Outline for Future Work:**

- Integration of External Data: Incorporating external datasets or sources of information, such as demographic or healthcare records, could enrich the dataset and improve model performance.
- Longitudinal Analysis: Conducting longitudinal studies to track maternal health outcomes over time could provide insights into disease progression and risk factors.
- Clinical Validation: Validating model predictions using clinical data and expert assessments could assess the real-world applicability and reliability of the models.
- Deployment and Implementation: Deploying models in clinical settings and integrating them into existing healthcare systems could facilitate early detection and intervention for maternal health risks.
- Continued Evaluation: Continuously evaluating model performance and updating models based on new data and emerging trends is crucial for maintaining effectiveness and relevance over time.

Conclusion

In this project, we developed a robust machine learning framework to classify maternal risk, employing a systematic methodology that included data exploration, preprocessing, feature engineering, and model evaluation, applying overall 3 experiments to handle duplicates and the unbalanced dataset all included models training and evaluation opting for a model that does not overfit nor underfit the data having good performance according to the used performance metrics .

The Random Forest model, validated through extensive evaluation metrics, demonstrated superior performance in classifying maternal risk. This model can assist healthcare professionals in identifying high-risk cases and making informed decisions, potentially improving maternal health outcomes. Future work could involve integrating additional data sources and exploring advanced machine learning techniques to further enhance model accuracy and reliability.

By leveraging this machine learning approach, the project aims to contribute to better healthcare management and improved maternal health, ultimately supporting efforts to reduce maternal mortality and morbidity.

References :

List all cited sources in a consistent citation style.

1. United Nations. (n.d.). Sustainable Development Goals: Goal 3. Retrieved from [UN Sustainable Development Goals - Goal 3]
2. Dataset Retrieved from [UC Irvine Machine Learning Repository]
3. https://www.researchgate.net/publication/374579002_AUTOMATED_CLASSIFICATION_OF_MATERNAL_RISKS_IN_PREGNANCY_ANALYSIS_USING_MACHINE_LEARNING_AND_ARTIFICIAL_NEURAL_NETWORKS
4. Maternal mortality (who.int)
5. Introductory paper Review and Analysis of Risk Factor of Maternal Health in Remote Area Using the Internet of Things (IoT)
6. MATERNAL RISK LEVEL PREDICTION USING ENSEMBLE MODEL
7. Machine Learning-based maternal health risk prediction model for IoMT framework