Maternal Health Risk Analysis Project Submission

This template helps you capture and share screenshots of your code output for project evaluation.

Instructions

You will submit nine screenshots as part of this final project.

To prepare your submission, follow these steps:

- Complete all the activities listed in each task below
- Take screenshots of the outputs mentioned in “**Screenshot(s) required for project evaluation**”
- Make sure each screenshot is clear and shows all the required information
- Insert your screenshots in the designated space within each table

Note: You will capture screenshots only for the key steps, but you must still complete all tasks to generate those screenshots.

Task 1: Prepare and clean the dataset

Activity number	Activity
1.	Import and clean maternal health data • Acquire and load the data • Remove duplicates • Remove any personally identifiable information (PII) • Handle missing values using median imputation • Normalize data values **Screenshot required for project evaluation** • Summary statistics (mean and standard deviation) of normalized data

Task 1: Summary statistics (mean and std) of normalized features

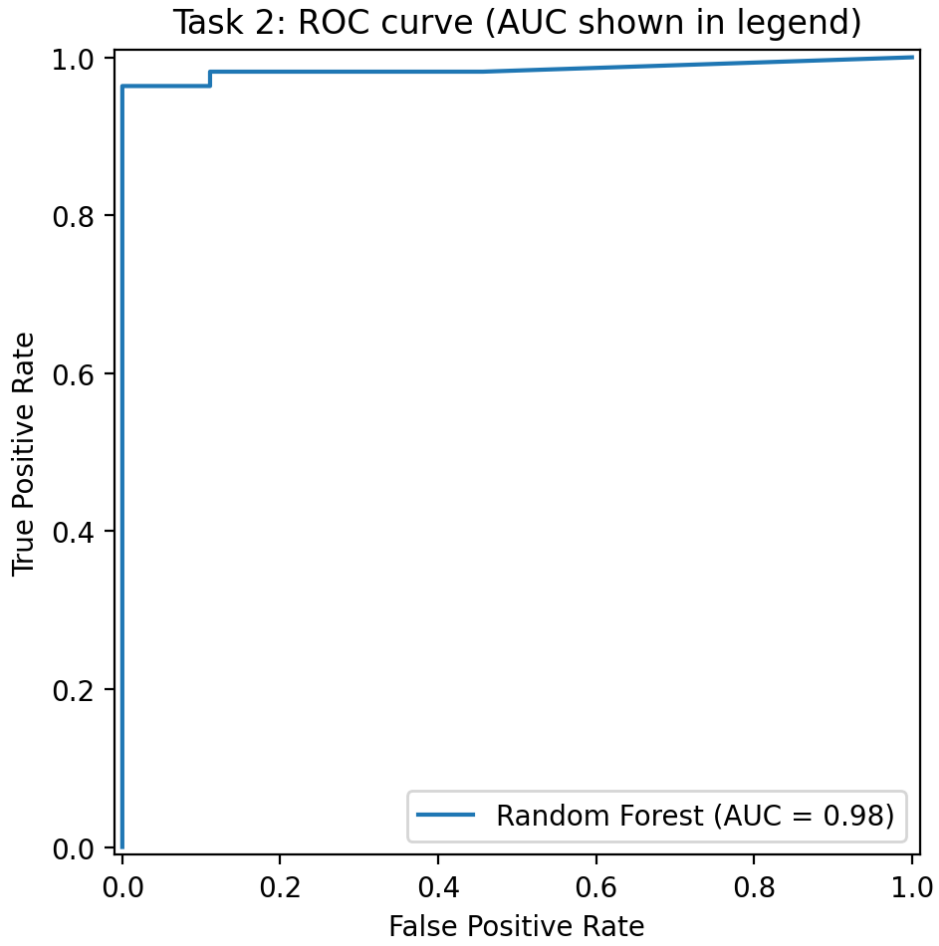
	mean	std
Age	0.3437	0.2308
SystolicBP	0.4802	0.2209
DiastolicBP	0.5601	0.2903
BS	0.2452	0.2772
BodyTemp	0.1163	0.2663
HeartRate	0.8116	0.1044

Task 2: Build and evaluate a supervised patient risk scoring model

Activity number	Activity
1.	Select features and create training–testing datasets • Identify the features (X) and target (y) columns • Perform an 80:20 train-test split, with stratification and random_state=42 **Screenshot required for project evaluation** • Target distribution for training and test sets (e.g., counts of each class) Task 2: Target distribution after 80:20 train-test split (stratified) Train (n) / Test (n) Low risk (0): 325 / 81 High risk (1): 217 / 55
2.	Implement and train a Random Forest classifier • Implement a Random Forest classifier with 100 trees and random_state=42 • Train the model and evaluate its performance using: ○ Accuracy

	○ Precision
	○ Recall
	○ F1-score
	
	Screenshot required for project evaluation
	• Single output showing all four metrics: Accuracy, Precision, Recall, and F1-score
	
	Task 2: Random Forest evaluation metrics
	`Random Forest (n_estimators=100, random_state=42)`
	`Test set metrics:`
	`  Accuracy : 0.9706`
	`  Precision: 0.9636`
	`  Recall   : 0.9636`
	`  F1-score : 0.9636`

3.	Visualize and save the trained model • Plot the ROC–AUC curve • Print feature importances • Save the trained model as risk_model.joblib **Screenshots required for project evaluation** • Screenshot 1: ROC-AUC curve (clearly labeled with axes and AUC score) • Screenshot 2: Top 3 feature importances with feature names and values


Task 2: ROC curve (AUC shown in legend)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Random Forest (AUC = 0.98)

	Task 2: Top 3 Random Forest feature importances |	importance
---	---
BS	0.4521
SystolicBP	0.2237
Age	0.097

Task 3: Develop and assess a dense neural network model

Activity number	Activity

1.	Scale data and train a Deep Neural Network
	
	• Scale the data using StandardScaler
	• Build and train a Deep Neural Network using:
	  ○ Loss: binary_crossentropy
	  ○ Optimizer: adam
	  ○ Epochs: 50
	
	Screenshot required for project evaluation
	• Summary statistics for a few representative features showing mean close to 0 and standard deviation close to 1 of scaled training data

Task 3: Scaled training data summary (mean≈0, std≈1)

	mean	std
Age	-0.0	1.0
SystolicBP	-0.0	1.0
DiastolicBP	0.0	1.0
BS	-0.0	1.0

2. Evaluate and save the Deep Neural Network model

- Evaluate the model using Accuracy, Precision, Recall, and F1-score on the test dataset
- Save the trained model as risk_model_dnn.keras

Screenshot required for project evaluation

- Evaluation metrics (Accuracy, Precision, Recall, F1-score) confirming model evaluation completed successfully

Task 3: DNN evaluation metrics

```
Dense Neural Network (binary_crossentropy, adam, epochs=50)
Test set metrics:
  Accuracy : 0.9632
  Precision: 0.9310
  Recall   : 0.9818
  F1-score : 0.9558
```

Task 4: Create and test a Gradio-based web application

Activity number	Activity

1.	Create the web application
	
	• Create a separate Python file named ai_risk_scoring_app.py
	• Load the saved Random Forest model (risk_model.joblib)
	• Build a Gradio-based web app that:
	  ○ Allows doctors to enter patient details (e.g., blood pressure, sugar, temperature, heart rate)
	  ○ Displays the predicted maternal risk level (e.g., “High Risk” or “Low Risk”)
	• Test and verify the web app functionality
	
	Screenshot required for project evaluation
	• Screenshot 1: Gradio interface with example patient values entered in all input fields
	• Screenshot 2: Prediction output showing the risk level
	
	Task 4: Gradio interface (replace with your screenshot)
	`Task 4 screenshot placeholder.`
	`Run the Gradio app and capture the interface with example values entered.`
	
	`Command:`
	`  python ai_risk_scoring_app.py`
	
	`Then open the local Gradio URL in your browser and take a screenshot.`

	Task 4: Prediction output (replace with your screenshot) Task 4 screenshot placeholder. After entering values, click 'Submit' and capture the prediction output showing 'High Risk' or 'Low Risk'.