Model Optimization and Tuning Phase Template

Date	13 June 2025
Name	Sakshi Santosh Patil
Project Title	Fetal Health Classification System
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase focuses on enhancing the performance of the XGBoost model to classify fetal health into Normal, Suspect, or Pathological categories using cardiotocography (CTG) datafromthefetal_health.csv dataset (2126 records, 80% training, 20% testing). This phase involves adjusting hyperparameters, evaluating performance metrics (accuracy, macro F1-score), and selecting the optimal model for deployment in the Fetal AI web application.

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters
Model1: XGBoost(Baseline)	- **n_estimators:** Number of boosting rounds. Tested value: [100]. Default value to establish a baseline. - **max_depth:** Maximumtree depth. Tested value: [6]. Balances complexity and overfitting. - **learning_rate:** Step size for updates. Tested value: [0.3]. Ensures faster initial convergence. - **subsample:** Fraction of samples per tree. Tested value: [1.0]. Uses all data for baseline training
Model2: XGBoost(Optimized)	- **n_estimators:** Increased rounds for better performance. Tested values: [100, 200, 300]. Selected: 200. - • **max_depth:** Adjusted for generalization. Tested values: [3, 6, 9]. Selected: 6. - • **learning_rate:** Lowered for stability. Tested values: [0.01, 0.1, 0.3]. Se lected: 0.1. - • **subsample:** Introduced randomness to reduce overfitting. Tested values: [0.7, 0.8, 1.0]. Selected: 0.8

Tuning Code Summary :
Hyperparameter tuning was implemented in train_model.py using GridSearchCV from Scikit-learn to optimize the XGBoost model.

```
from sklearn.model_selection import GridSearchCV
from xgboost import XGBClassifier
import joblib

# Define parameter grid
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [3, 6, 9],
    'learning_rate': [0.01, 0.1, 0.3],
    'subsample': [0.7, 0.8, 1.0]
}

# Perform grid search
grid_search = GridSearchCV(
    XGBClassifier(use_label_encoder=False, eval_metric='mlogloss',
       random_state=42),
    param_grid, cv=5, scoring='f1_macro', n_jobs=-1
)
grid_search.fit(X_train, y_train)

# Save optimized model
joblib.dump(grid_search.best_estimator_, 'fetalai_model.pkl')
```

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
XGBoost (Optimized)	The optimized XGBoost model was selected for Fetal AI due to its improved performance over the baseline model. With hyperparameters set to n_estimators=200, max_depth=6, learning_rate=0.1, and subsample=0.8, it achieved a test ac curacy of ~90% and a macro F1-score of ~0.85, compared to the baseline's 88% accuracyand0.82F1-score. The optimized model better handles class imbalance (Normal, Suspect, Pathological) and generalizes well to unseen data, critical for 2 medical applications. The tuning process improved precision and recall for mi nority classes, justifying the computational cost and making the model suitable for deployment in the Fetal AI Flask application