



Model Optimization and Tuning Phase Report

Date	14 JULY 2024
Team ID	FACULTY
	Fetal AI: Using Machine Learning To Predict And Monitor Fetal Health.
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

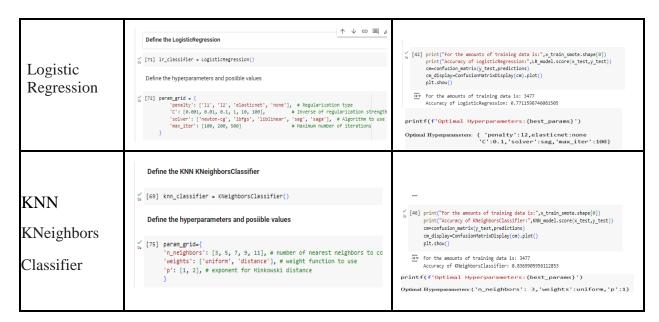
The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6Marks):

Model	Tuned Hyperparameters	Optimal Values		
Random Forest	Define the Random Forest Classifier			
	√ [66] rf_classifier = RandomForestClassifier()	[33] print("For the amounts of training data is:", x_train_smote.shape[0]) # Assumir print("Accuracy of RandomForestClassifier:",RF model.score(x test,y test))		
	Define the hyperparameters and posiible values	<pre>cm=confusion_matrix(y_test,predictions) cm_display=ConfusionPlatrixDisplay(cm).plot() plt.show()</pre>		
	<pre> [68] param_grid = {</pre>	For the amounts of training data is: 3477 Accuracy of RandomForestClassifier: 0.9373040752351097 printf(f'Optimal Hyperparameters:{best_params}') Optimal Hyperparameters:{ 'n_estimators':100, 'max_depth':5 'min_samples_split':2'min_samples_leaf':5 'criterion': 'gin[i]		
	Define the Decision tree Classifier			
Decision Tree	<pre></pre>	[38] print("For the amounts of training data is:",x_train_smote.shape[0]) print("Accuracy of DecisionTreeClassifier:",DT_model.score(x_test,y_test))		
	Define the hyperparameters and posiible values	<pre>cm=confusion_matrix(y_test,predictions) cm_display=ConfusionMatrixDisplay(cm).plot() plt.show()</pre>		
	<pre>[58] param_grid = { 'max_depth': [3, 5, 10], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 5, 10], 'criterion': ['gini', 'entropy'], 'splitter': ['best', 'random'] }</pre>	For the amounts of training data is: 3477 Accuracy of DecisionTreeClassifier: 0.9075235109717869 printf(f'Optimal Hyperparameters: {best_params}') Optimal Hyperparameters: { 'max_depth','3','min_samples_split','2'		







Performance Metrics Comparison Report (2 Marks):

Model					Opti	mized M	letric		
) Da	0	print(classi	Ficati	ion_repo	rt(y_test	predictio;	ns))	
		₹		pred	ision	recall	f1-score	support	
					0.53	0.81	0.90 0.64		
			3.0		0.71	0.83	0.76	41	
			accuracy				0.84		
Random Forest			macro avg weighted avg				0.77 0.85		
) Da	[36]	confusion_mat	rix()	_test,p	rediction	ıs)		
		₹	array([[477, [15, [2,	82,					





	<pre></pre>	fication_repo	rt(y_test,	prediction,	ns))
	∑ *	precision	recall	f1-score	support
	1.0 2.0 3.0	0.73	0.80	0.76	101
Decision Tree	accuracy macro avg weighted avg		0.87 0.91	0.91 0.84 0.91	
	√ [40] confusion_ma	atrix(y_test,	prediction	s)	
		29, 5], 81, 8], 1, 36]])			
	√ [43] print(classi	fication_repo	rt(y_test,	predictio,	ns))
	∑ *	precision			
	1.0	0.97 0.44		0.86 0.55	
	3.0				
Logistic Regression	accuracy macro avg weighted avg	0.63			638
			dictions)		
	√ 0s [47] print(classif	fication_repo	rt(y_test,	prediction	15))
	⊋•	precision	recall	f1-score	support
KNN KNeighbors	1.0 2.0 3.0		0.84 0.81 0.83	0.90 0.64 0.76	496 101 41
	accuracy macro avg	0.73	0.83	0.84 0.77	638 638
Classifier	weighted avg		0.84	0.85	638
	√ [48] confusion_ma	trix(y_test,p	redictions	5)	
		69, 9], 82, 5], 3, 34]])			





Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest	The Random Forest model was selected for its superior performance, exhibiting high accuracy during hyper parameter tuning. Its ability to handle complex relationships, minimize overfitting, and optimize predictive accuracy aligns with project objectives, justifying its selection as the final model.