Model Optimization and Tuning Phase

Date	19 June 2025
Team ID	SWTID17449620488
Project Title	Early Prediction for Chronic Kidney Disease Detection: A Progressive Approach to Health Management
Maximum Marks	10 Marks

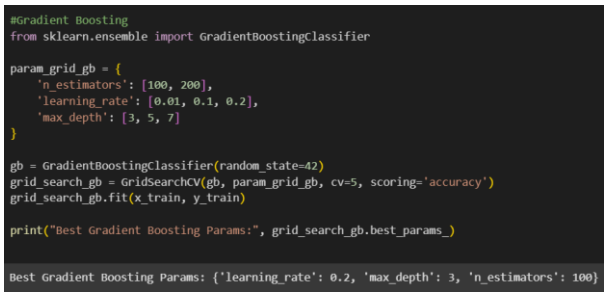
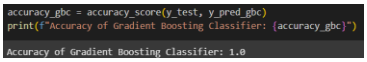
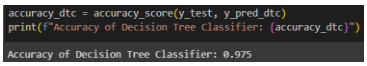
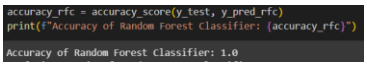
Model Optimization and Tuning Phase

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 (6 Marks):

Model	Tuned Hyperparameters	Optimal Values

Logistic Regression	`#Logistic Regression`
	`from sklearn.linear_model import LogisticRegression`
	`from sklearn.model_selection import GridSearchCV`
	`import warnings`
	`from sklearn.exceptions import ConvergenceWarning`
	
	`# Suppress convergence warnings`
	`warnings.filterwarnings("ignore", category=ConvergenceWarning)`
	
	`param_grid_lr = {`
	`'penalty': ['l1', 'l2'],`
	`'C': [0.01, 0.1, 1, 10],`
	`'solver': ['liblinear', 'saga']`
	`}`
	
	`lr = LogisticRegression(max_iter=2000)`
	`grid_search_lr = GridSearchCV(lr, param_grid_lr, cv=5, scoring='accuracy')`
	`grid_search_lr.fit(x_train, y_train)`
	
	`print("Best Logistic Regression Params:", grid_search_lr.best_params_)`
	
	`Best Logistic Regression Params: {'C': 1, 'penalty': 'l2', 'solver': 'liblinear'}`
		`accuracy = accuracy_score(y_test, y_pred)`
		`print(f"Accuracy: {accuracy}")`
		
		`Accuracy: 0.925`

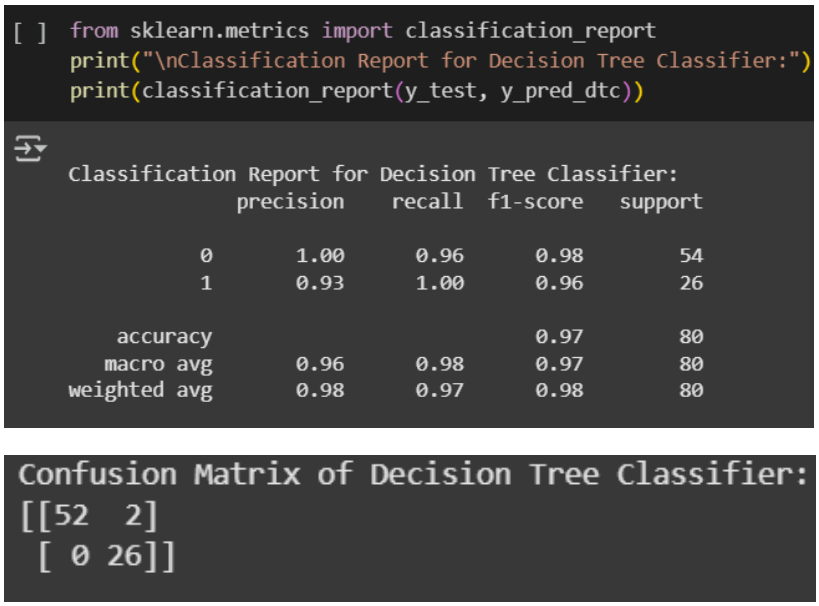
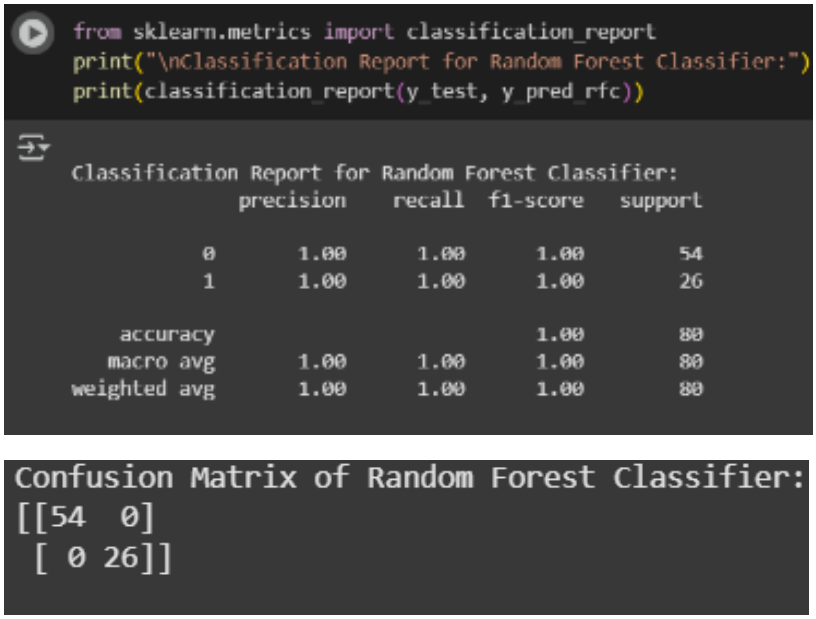
Gradient Boosting		
Decision Tree		
Random Forest		

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric

Logistic Regression	`from sklearn.metrics import classification_report`
	`print("Classification Report for Logistic Regression:")`
	`print(classification_report(y_test, y_pred))`
	
	Classification Report for Logistic Regression:
	precision recall f1-score support
	
	0 0.94 0.94 0.94 54
	1 0.88 0.88 0.88 26
	
	accuracy 0.93 80
	macro avg 0.91 0.91 0.91 80
	weighted avg 0.93 0.93 0.93 80
	
	Confusion Matrix:
	[[51 3]
	[3 23]]

Gradient Boosting	`from sklearn.metrics import classification_report`
	`print("\nClassification Report for Gradient Boosting Classifier:")`
	`print(classification_report(y_test, y_pred_gbc))`
	
	Classification Report for Gradient Boosting Classifier:
	precision recall f1-score support
	
	0 1.00 1.00 1.00 54
	1 1.00 1.00 1.00 26
	
	accuracy 1.00 80
	macro avg 1.00 1.00 1.00 80
	weighted avg 1.00 1.00 1.00 80
	
	Confusion Matrix of Gradient Boosting Classifier:
	[[54 0]
	[0 26]]

Decision Tree	
Random Forest	

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest	It achieved the highest validation accuracy among all tuned models. It also handles high-dimensional data well, reduces overfitting through bagging, and was optimized using hyperparameter tuning.