



## **Model Optimization and Tuning Phase Template**

Date	14 July 2024
Team ID	SWTID1720190579
Project Title	Early Prediction of Chronic Kidney Disease Using Machine Learning
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
KNN	<pre>from sklearn.neighbors import KNeighborsClassifier knn = KNeighborsClassifier(n_neighbors=6, weights='uniform', algorithm='kd_tree', leaf_size=20) knn.fit(x_train, y_train)</pre>	<pre>accuracy_score(y_pred, y_test) 0.9625</pre>
Logistic Regression	[63]: from sklearn.linear_model import LogisticRegression  lr = LogisticRegression(random_state=42)  lr.fit(X_train, y_train)  [63]: LogisticRegression  LogisticRegression(random_state=42)	<pre>lr_acc = accuracy_score(y_pred, y_test) lr_acc 0.9625</pre>

## **Performance Metrics Comparison Report (2 Marks):**

Model	Base line Metric/Optimized Metric





KNN	# Print the classification report and confusion matrix print("Confusion Matrix:") print(confusion_matrix(y_test, pred)) print("Colassification_report(y_test, pred))  Accuracy: 0.975 Confusion Matrix: [[\$1 1] [ 1 27]]  Classification recall f1-score support  0 0.98 0.98 0.98 52 1 0.96 0.96 0.96 28  accuracy 0.97 80 macro avg 0.97 0.97 0.97 80 weighted avg 0.97 0.97 0.97 80					
	from sklearm.metrics import accuracy_score accuracy_score(y_test.pred) print("confusion Matrix:") print(confusion_matrix:") print("valussification.Report:") print("valussification.Report(y_test, pred)) print(classification_report(y_test, pred)) print(classification_report(y_test, pred))					
	0 0.98 0.92 0.95 52 1 0.87 0.96 0.92 28					
	accuracy 0.94 80					
Naive Bayes	macro avg 0.93 0.94 0.93 80 weighted avg 0.94 0.94 0.94 80 Confusion Matrix:					
	Confusion Matrix: [[48 4] [ 1 27]]					
	Classification Report: precision recall f1-score support					
	0 0.98 0.92 0.95 52 1 0.87 0.96 0.92 28					
	accuracy 0.94 80 macro avg 0.93 0.94 0.93 80					
	macro avg 0.93 0.94 0.93 80 weighted avg 0.94 0.94 0.94 80					
	# Generate a classification report report = classification_report(y_test, y_pred) print("Classification Report:") print("confusion Matrix:") print("confusion Matrix:") print("confusion_matrix(y_test, pred)) print("nclassification_Report:") print(classification_report(y_test, pred)) Accuracy: 0.975 Classification_Report:					
	precision recall f1-score support					
	0 0.98 0.98 0.98 52 1 0.96 0.96 0.96 28					
SVM	accuracy 0.97 80 macro avg 0.97 0.97 80 weighted avg 0.97 0.97 0.97 80					
	Confusion Matrix: [[33 19] [16 12]]					
	Classification Report: precision recall f1-score support					
	0 0.67 0.63 0.65 52 1 0.39 0.43 0.41 28					
	accuracy 0.56 80					
	macro avg 0.53 0.53 0.53 80 weighted avg 0.57 0.56 0.57 80					





	print("Confusi print(confusio print("\nClass	n_matrix(	y_test, p			
	<pre>print(classification_report(y_test, pred))</pre>					
	Accuracy: 0.97					
	Classification	Report: precision	recal	l f1-scor	support	
	0	0.98				
	accuracy macro avg	0.97	0.9	7 0.9		
Logistic Regression	weighted avg	0.97				
8	Confusion Matr [[33 19] [16 12]]	ix:				
	Classification	Report:				
		precision	recal:	l f1-scor	support	
	9	0.67	0.6	9.6	52	
	1	0.39				
	accuracy			0.5	80	
	macro avg	0.53		0.5	80	
	weighted avg	0.57	0.5	6 0.5	88	
	print("Confusion			))		
	<pre>print("Confusion print(confusion_ print("\nClassifi print(classificat</pre>	matrix(y_t cation Re	est, pred port:")			
	<pre>print(confusion_n print("\nClassifi print(classificat  Accuracy: 0.95</pre>	matrix(y_t cation Re cion_repor	est, pred port:")			
	print(confusion_n print("\nClassifi print(classificat Accuracy: 0.95 Classification Re	matrix(y_t cation Re dion_report	est, pred port:") t(y_test,	pred))	support	
	print(confusion_n print("\nClassifis print(classificat Accuracy: 0.95 Classification Re	matrix(y_t cation Re cion_repor	est, pred port:") t(y_test,	pred)) f1-score	support	
	print(confusion_n print("\nClassifi print(classificat Accuracy: 0.95 Classification Re	matrix(y_t cation Re dion_report	est, pred port:") t(y_test,	pred))	support 52 28	
	print(confusion_n print("\nClassifi print(classificat Accuracy: 0.95 Classification Re pre	matrix(y_t cation Re cion_repor eport: ecision 0.98	est, pred port:") t(y_test,  recall 0.94	pred))  f1-score  0.96  0.93	52 28	
	print(confusion_n print("\nclassific print(classification Accuracy: 0.95 Classification Re pre	matrix(y_t cation Re cion_repor eport: ecision 0.98	est, pred port:") t(y_test,  recall 0.94	pred)) f1-score 0.96	52	
Decision Tree	print(confusion_n print("\nClassifi print(classificat Accuracy: 0.95 Classification Re pre	eatrix(y_t cation Re cion_repor eport: ecision 0.98 0.90	est, pred port:") t(y_test, recall 0.94 0.96	pred))  f1-score  0.96  0.93  0.95	52 28	
Decision Tree	print(confusion_n print("Nclassificat print(classificat Accuracy: 0.95 Classification Re pre 0 1 accuracy macro avg	eport: cision  0.98 0.99  0.94 0.95	est, pred port:") t(y_test,  recall 0.94 0.96	pred))  f1-score  0.96  0.93  0.95  0.95	52 28 80 80	
Decision Tree	print(confusion, aprint("NCLassification print(classification Recuracy: 0.95 Classification Recuracy: 0.95 Classification Recuracy macro avg weighted avg Confusion Matrix: [[33 19] [16 12]] Classification Recuracy macro avg weighted avg Confusion Matrix: [13 19] [16 12]] Classification Recuracy macro avg weighted avg Matrix: [13 19] [16 12]] Classification Recuracy macro avg weighted avg weig	matrix(y_t cation Re cion_report eport: ccision 0.98 0.90 0.94 0.95	est, pred port:") t(y_test, recall : 0.94 0.96 0.95 0.95	pred))  f1-score      0.96     0.93     0.95     0.95	52 28 80 80	
Decision Tree	print(confusion, aprint("NCLassification print(classification Recuracy: 0.95 Classification Recuracy: 0.95 Classification Recuracy macro avg weighted avg Confusion Matrix: [[33 19] [16 12]] Classification Recuracy macro avg weighted avg Confusion Matrix: [13 19] [16 12]] Classification Recuracy macro avg weighted avg Matrix: [13 19] [16 12]] Classification Recuracy macro avg weighted avg weig	eport: ccision 0.98 0.99 0.94 0.95	est, pred port:") t(y_test,  recall 0.94 0.96	pred))  f1-score      0.96     0.93     0.95     0.95	52 28 80 80	
Decision Tree	print(confusion, aprint("NCLassification print(classification Recuracy: 0.95 Classification Recuracy: 0.95 Classification Recuracy macro avg weighted avg Confusion Matrix: [[33 19] [16 12]] Classification Recuracy macro avg weighted avg Confusion Matrix: [13 19] [16 12]] Classification Recuracy macro avg weighted avg Matrix: [13 19] [16 12]] Classification Recuracy macro avg weighted avg weig	matrix(y_t cation Re cion_report eport: ccision 0.98 0.90 0.94 0.95	est, pred port:") t(y_test, recall : 0.94 0.96 0.95 0.95	pred))  f1-score      0.96     0.93     0.95     0.95	52 28 89 80 80	
Decision Tree	print(confusion, aprint("NCLassification Recuracy: 0.95 Classification Recuracy: 0.95 Classification Recuracy macro avg weighted avg elegated avg (Confusion Matrix: [33 19] [16 12]] Classification Recuracy pre-	eatrix(y_t cation Re cion_report eport: ecision 0.98 0.90 0.94 0.95	est, pred port:") t(y_test,  recall 0.94 0.96 0.95 0.95	f1-score 0.96 0.93 0.95 0.95 0.95	52 28 80 80 80	
Decision Tree	print(confusion a print('nClassifi print(classification Recursey) 0.95 classification Recursey macro avg weighted avg Confusion Matrix: [[33 19] [16 12]] classification Recursey pre confusion Matrix: [13 19] [16 12]] classification Recursey pre confusion Matrix: [13 19] [16 12]] classification Recursey pre confusion Matrix: [13 19] [16 12]] classification Recursey pre confusion Matrix: [18 18 18 18 18 18 18 18 18 18 18 18 18 1	matrix(y_t cation Re cion_report  port: cision  0.98  0.99  0.94  0.95	est, pred port:") t(y_test, recall 0.94 0.96 0.95 0.95	f1-score 0.96 0.93 0.95 0.95 0.95	52 28 80 80 80 80	
Decision Tree	print(confusion, print("nclassification Recuracy: 0.95 classification Recuracy macro avg weighted avg Confusion Matrix: [[33 19] [16 12]] classification Recuracy macro avg avg confusion Matrix: [[33 19] [16 12]] classification Recuracy macro avg macro avg macro avg	eport: cision 0.98 0.90 0.94 0.95	est, pred port:") t(y_test, t(y_test, t(y_test), t(y_te	f1-score 0.96 0.93 0.95 0.95 0.95 1-score 0.65 0.41 0.56 0.53	52 28 80 80 80 80 80 52 28 80 80	
Decision Tree	print(confusion, aprint("NCLassification print("Classification Recuracy: 0.95 Classification Recuracy macro avg weighted avg weighted avg (Confusion Matrix: [33 19] [16 12]] Classification Recuracy according to the precuracy for	patrix(y_t cation Re cion_report  port: cision  0.98 0.90  0.94 0.95	est, pred port:") t(y_test, t(y_test, 0.94 0.96 0.95 0.95 0.95 0.95	f1-score 0.96 0.93 0.95 0.95 0.95 1-score 0.65 0.41	52 28 80 80 80 80 support	
Decision Tree	print(confusion, print("nclassification Recuracy: 0.95 classification Recuracy macro avg weighted avg Confusion Matrix: [[33 19] [16 12]] classification Recuracy macro avg avg confusion Matrix: [[33 19] [16 12]] classification Recuracy macro avg macro avg macro avg	eport: cision 0.98 0.90 0.94 0.95	est, pred port:") t(y_test, t(y_test, t(y_test), t(y_te	f1-score 0.96 0.93 0.95 0.95 0.95 1-score 0.65 0.41 0.56 0.53	52 28 80 80 80 80 80 52 28 80 80	
Decision Tree	print(confusion, print("nclassification Recuracy: 0.95 classification Recuracy macro avg weighted avg Confusion Matrix: [[33 19] [16 12]] classification Recuracy macro avg avg confusion Matrix: [[33 19] [16 12]] classification Recuracy macro avg macro avg macro avg	eport: cision 0.98 0.90 0.94 0.95	est, pred port:") t(y_test, t(y_test, t(y_test), t(y_te	f1-score 0.96 0.93 0.95 0.95 0.95 1-score 0.65 0.41 0.56 0.53	52 28 80 80 80 80 80 52 28 80 80	

# **Final Model Selection Justification (2 Marks):**

Final Model	Reasoning
	The logistic regression model was selected for its superior performance,
	exhibiting high accuracy during hyperparameter tuning. Its ability to handle
	complex relationships, minimize overfitting, and optimize predictive accuracy
	aligns with project objectives, justifying its selection as the final model. The
	Logistic Regression model provides detailed metrics showing an accuracy of
Logistic Regression	0.66 and a weighted average F1-score of 0.61.