



Model Optimization and Tuning Phase Template

Date	10 JULY 2024
Team ID	SWTID1720193784
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	In [65]: from sklearn.neighbors import NNeighborsClassifier knn = KNeighborsClassifier(n_neighbors=6, weights='uniform', algorithm='kd_tree', leaf_size=20) In [66]: knn.fit(x_train, y_train) Out[66]:	In [67]: accuracy_score(y_pred, y_test) Out[67]: 0.9625
LOGISTIC REGRESSION	<pre>In [63]: from sklearn.linear_model import LogisticRegression</pre>	In [64]: lr_acc = accuracy_score(y_pred, y_test) lr_acc Out[64]: 0.9625





Performance Metrics Comparison Report (2 Marks):

Model		Optim	ized Metric	2	
	<pre># Calculate the accuracy = accu print(f"Accuracy # Print the class print("Confusion print(confusion print("\nClassi print(classifice)</pre>	racy_score y: {accura ssificatio n Matrix:" _matrix(y_ fication R	<pre>cy}") n report) test, pre eport:")</pre>	and confusi	ion matrix
KNN	Accuracy: 0.987 Confusion Matri [[52 0] [1 27]]				
	Classification p	Report: recision	recall	f1-score	support
	0 1	0.98 1.00	1.00 0.96		52 28
	accuracy macro avg weighted avg	0.99 0.99	0.98 0.99		80 80 80





	<pre># Calculate the accuracy score accuracy = accuracy_score(y_test, y_pred) print("Accuracy:", accuracy) # Generate a classification report report = classification_report(y_test, y_pred) print("Classification Report:") print(report) print("Confusion Matrix:") print(confusion_matrix(y_test, pred)) print("\nClassification Report:") print(classification_report(y_test, pred))</pre>					
	Accuracy: Classific		eport:			
		pre	cision	recall	f1-score	support
		0	0.98	0.98	0.98	52
SVM		1	0.96	0.96	0.96	28
5 7 171	accur	acv			0.97	80
	macro		0.97	0.97	0.97	80
	weighted	avg	0.97	0.97	0.97	80
	Confusion [[36 16] [18 10]]					
	Classific	ation Re	port:			
			cision	recall	f1-score	support
		0	0.67	0.69	0.68	52
		1	0.38	0.36	0.37	28
	accur	acy			0.57	80
	macro		0.53	0.52	0.52	80
	weighted	avg	0.57	0.57	0.57	80





	<pre>print("Confusion Matrix:") print(confusion_matrix(y_test, pred)) print("\nClassification Report:") print(classification_report(y_test, pred))</pre>					
	Accuracy: 0.975 Classification Report: precision recall f1-score suppor					
	0 1	0.98 0.96	0.98 0.96	0.98 0.96	52 28	
LOGISTIC REGRESSION	accuracy macro avg weighted avg	0.97 0.97	0.97 0.97		80 80 80	
	Confusion Matrix: [[36 16] [18 10]]					
	Classification		11	£1		
		precision		f1–score	support	
	0	0.67 0.38	0.69 0.36	0.68 0.37	52 28	
	accuracy macro avg	0.53	0.52	0.57 0.52	80 80	
	weighted avg	0.57	0.57		80	





	<pre>from sklearn.metrics import accuracy_score accuracy_score(y_test,pred) print("Confusion Matrix:") print(confusion_matrix(y_test, pred)) print("\nClassification Report:") print(classification_report(y_test, pred))</pre>					
		precision	recall	f1-score	support	
	0 1	0.96 1.00	1.00 0.93	0.98 0.96	52 28	
NAIVE BAYES	accuracy macro avg weighted avg	0.98 0.98	0.96 0.97	0.97 0.97 0.97	80 80 80	
	Confusion Mat [[52 0] [2 26]]	rix:				
	Classificatio	n Report:				
		precision	recall	f1-score	support	
	0 1	0.96 1.00	1.00 0.93	0.98 0.96	52 28	
	accuracy			0.97	80	
	macro avg weighted avg	0.98 0.98	0.96 0.97	0.97 0.97	80 80	

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Gradient Boosting	The Gradient Boosting 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.