Model Optimization and Tuning Phase Template

Date	15 March 2024
Team ID	xxxxxx
Project Title	xxxxxx
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

Random forest	```# Define the Decision Tree classifier
	dt_classifier = DecisionTreeClassifier()
	
	# Define the hyperparameters and their possible values for tuning
	param_grid = {
	    'criterion': ['gini', 'entropy'],
	    'splitter': ['best', 'random'],
	    'max_depth': [None, 10, 20, 30, 40, 50],
	    'min_samples_split': [2, 5, 10],
	    'min_samples_leaf': [1, 2, 4]
	}```
		# Evaluate the performance of the tuned model
		accuracy = accuracy_score(y_test, y_pred)
		print(f"Optimal Hyperparameters: {best_params}")
		print(f"Accuracy on Test Set: {accuracy}")
		
		Optimal Hyperparameters: {'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter':
		Accuracy on Test Set: 0.7297931369847

Decision tree	```# Define the Random Forest classifier
	rf_classifier = RandomForestClassifier()
	
	# Define the hyperparameters and their possible values for tuning
	param_grid = {
	    'n_estimators': [50, 100, 200],
	    'criterion': ['gini', 'entropy'],
	    'max_depth': [None, 10, 20, 30],
	    'min_samples_split': [2, 5, 10],
	    'min_samples_leaf': [1, 2, 4],
	}```
		# Evaluate the performance of the tuned model
		accuracy = accuracy_score(y_test, y_pred)
		print(f"Optimal Hyperparameters: {best_params}")
		print(f"Accuracy on Test Set: {accuracy}")
		
		Optimal Hyperparameters: {'criterion': 'entropy', 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators'
		Accuracy on Test Set: 0.7710470994918

...	...	...

Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric

Decision tree	…	```print(classification_report(y_test,y_pred))```
		
		` ` precision recall f1-score support
		
		`    Loan will be Approved` 0.67 0.68 0.68 75
		`Loan will not be Approved` 0.74 0.73 0.74 94
		
		`             accuracy` 0.71 169
		`            macro avg` 0.71 0.71 0.71 169
		`         weighted avg` 0.71 0.71 0.71 169
		
		```confusion_matrix(y_test,y_pred)```
		```array([[51, 24],```
		`       [25, 69]])`

Random forest	…	```print(classification_report(y_test,y_pred))```
		
		` ` precision recall f1-score support
		
		`    Loan will be Approved` 0.71 0.83 0.77 75
		`Loan will not be Approved` 0.84 0.73 0.78 94
		
		`             accuracy` 0.78 169
		`            macro avg` 0.78 0.78 0.77 169
		`         weighted avg` 0.78 0.78 0.78 169
		
		```confusion_matrix(y_test,y_pred)```
		```array([[62, 13],```
		`       [25, 69]])`

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Model 1 (or other)	Explanation of why this model was chosen as the final optimized model