Model Optimization and Tuning Phase Template

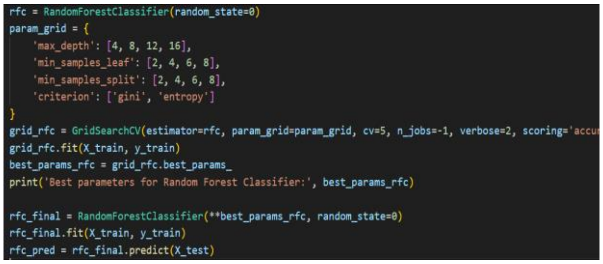
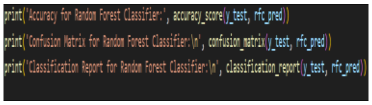
Date	15 March 2024
Team ID	SWTID1720452383
Project Title	Ecommerce Shipping Prediction Using Machine
Maximum Marks	10 Marks

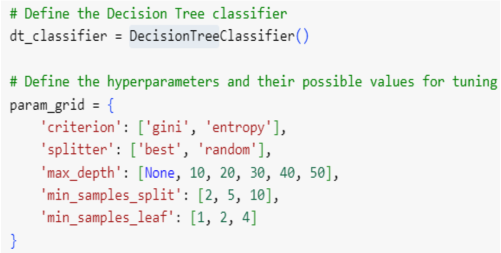
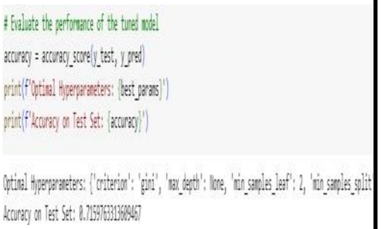
Model Optimization and Tuning Phase

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):

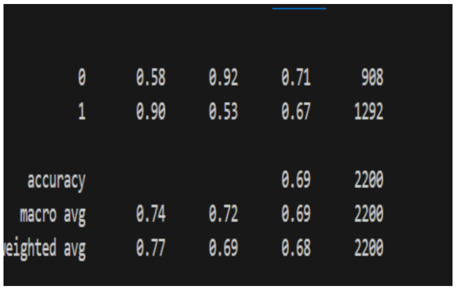
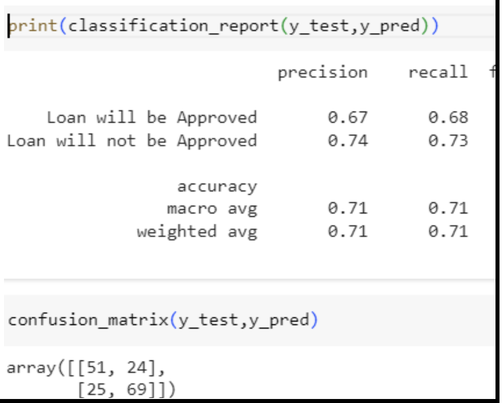
Model	Tuned Hyperparameters	Optimal Values

Random forest classifire	
	rfc = RandomForestClassifier(random_state=0)
	param_grid = {
	    'max_depth': [4, 8, 12, 16],
	    'min_samples_leaf': [2, 4, 6, 8],
	    'min_samples_split': [2, 4, 6, 8],
	    'criterion': ['gini', 'entropy']
	}
	grid_rfc = GridSearchCV(estimator=rfc, param_grid=param_grid, cv=5, n_jobs=-1, verbose=2, scoring='accur
	grid_rfc.fit(X_train, y_train)
	best_params_rfc = grid_rfc.best_params_
	print('Best parameters for Random Forest Classifier:', best_params_rfc)
	
	rfc_final = RandomForestClassifier(**best_params_rfc, random_state=0)
	rfc_final.fit(X_train, y_train)
	rfc_pred = rfc_final.predict(X_test)
		
		print('Accuracy for Random Forest Classifier:', accuracy_score(y_test, rfc_pred))
		print('Confusion Matrix for Random Forest Classifier:\n', confusion_matrix(y_test, rfc_pred))
		print('Classification Report for Random Forest Classifier:\n', classification_report(y_test, rfc_pred))

Decision Tree	
	# 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': 2, 'min_samples_split'
		Accuracy on Test Set: 0.7159763313609467

…	…	…

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric
Random Forest classifier	 0 0.58 0.92 0.71 908 1 0.90 0.53 0.67 1292 accuracy 0.69 2200 macro avg 0.74 0.72 0.69 2200 eighted avg 0.77 0.69 0.68 2200
Decision Tree	 print(classification_report(y_test,y_pred)) precision recall f Loan will be Approved 0.67 0.68 Loan will not be Approved 0.74 0.73 accuracy macro avg 0.71 0.71 weighted avg 0.71 0.71 confusion_matrix(y_test,y_pred) array([[51, 24], [25, 69]])

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Randomforest classifier	The random foest classifier was choosen as the final model as this model provides us with a higher percentage of accuracy with a highly optimizd code and better runtime