Model Optimization and Tuning Phase Report

Date	23 September 2024
Team ID	LTVIP2024TMID24967
Project Title	SmartLender - Applicant Credibility Prediction for Loan Approval
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

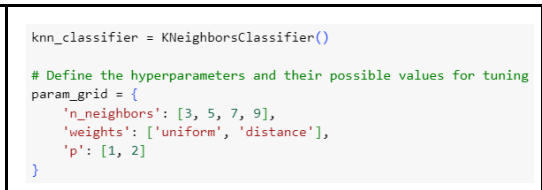
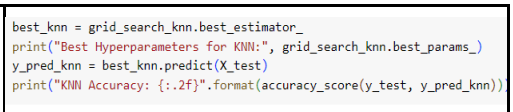
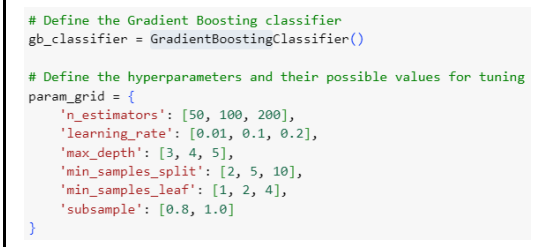
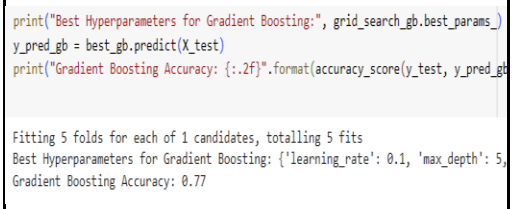
The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values

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]`
	`}`
		`best_dt = grid_search_dt.best_estimator_`
		`print("Best Hyperparameters for Decision Tree:", grid_search_dt.best_params_)`
		`y_pred_dt = best_dt.predict(X_test)`
		`print("Decision Tree Accuracy: {:.2f}".format(accuracy_score(y_test, y_pred_dt)))`
		
		`Fitting 5 folds for each of 216 candidates, totalling 1080 fits`
		`Best Hyperparameters for Decision Tree: {'criterion': 'gini', 'max_depth': 10, 'max_`
		`Decision Tree Accuracy: 0.74`

Random Forest	`# 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],`
	`}`
		`best_rf = grid_search_rf.best_estimator_`
		`print("Best Hyperparameters for Random Forest:", grid_search_rf.best_params_)`
		`y_pred_rf = best_rf.predict(X_test)`
		`print("Random Forest Accuracy: {:.2f}".format(accuracy_score(y_test, y_pred_rf)))`
		
		`Fitting 5 folds for each of 216 candidates, totalling 1080 fits`
		`Best Hyperparameters for Random Forest: {'bootstrap': True, 'max_depth': 30, 'min_samp`
		`Random Forest Accuracy: 0.80`

KNN		
Gradient Boosting		

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric
Decision Tree	**Decision Tree Classifier** Confusion matrix [[54 20] [13 57]] Classification report precision recall f1-score support 0 0.81 0.73 0.77 74 1 0.74 0.81 0.78 70 accuracy 0.77 144 macro avg 0.77 0.77 0.77 144 weighted avg 0.77 0.77 0.77 144

Random Forest	*** Random Forest Classifier *** Confusion matrix: [[54 20] [5 65]] Classification report: precision recall f1-score support 0 0.92 0.73 0.81 74 1 0.76 0.93 0.84 70 accuracy 0.83 144 macro avg 0.84 0.83 0.83 144 weighted avg 0.84 0.83 0.82 144
KNN	*** KNeighbors Classifier *** Confusion matrix: [[45 29] [7 63]] Classification report: precision recall f1-score support 0 0.87 0.61 0.71 74 1 0.68 0.90 0.78 70 accuracy 0.75 144 macro avg 0.78 0.75 0.75 144 weighted avg 0.78 0.75 0.75 144
Gradient Boosting	*** Gradient Boosting Classifier *** Confusion matrix: [[44 30] [5 65]] Classification report: precision recall f1-score support 0 0.90 0.59 0.72 74 1 0.68 0.93 0.79 70 accuracy 0.76 144 macro avg 0.79 0.76 0.75 144 weighted avg 0.79 0.76 0.75 144

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest	The Random Forest model was selected for its robust performance and strong generalization capabilities. Its ability to handle a large number of features, mitigate overfitting through bagging, and provide feature importance metrics made it a top contender. Random Forest demonstrated competitive accuracy during hyperparameter tuning, and its ensemble nature ensures better stability and resilience to noise in the dataset. These factors align with the project's objectives, making it an excellent choice as the final model.