Model Optimization and Tuning Phase Template

Date	**11 July 2024**
Team ID	**SWTID1720115788**
Project Title	**Ecommerce Shipping Prediction Using Machine Learning**
Maximum Marks	**10 Marks**

Model Optimization and Tuning Phase

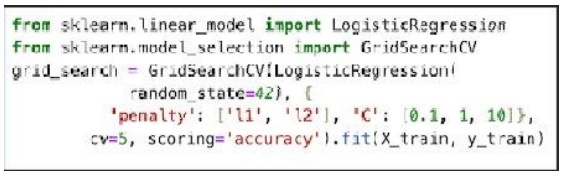
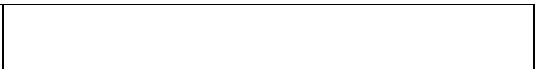
The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (6 Marks):

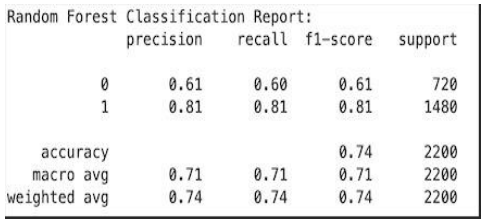
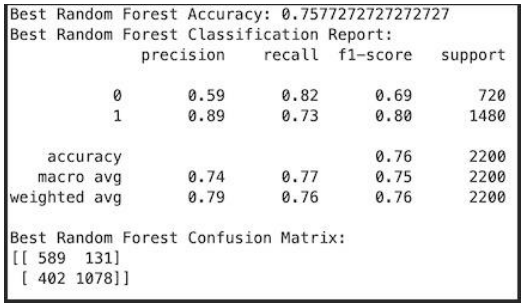
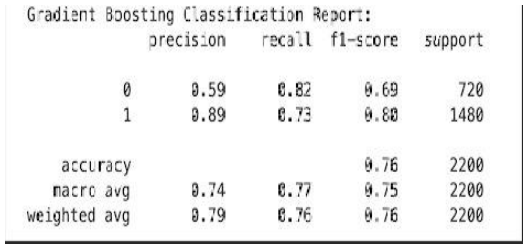
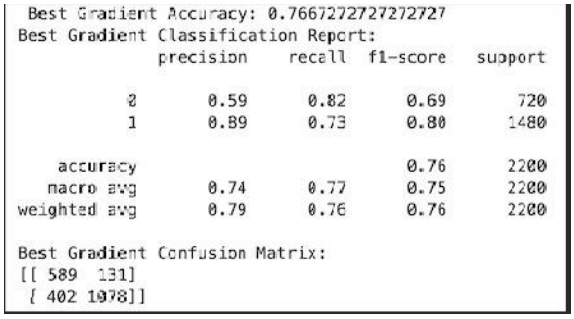
Model	Tuned Hyperparameters	Optimal Values
Random Forest	```## Tuned HyperParameters from sklearn.ensemble import RandomForestClassifier from sklearn.model_selection import GridSearchCV from sklearn.metrics import accuracy_score, classification_report, confusion_matrix rf = RandomForestClassifier(criterion='entropy',random_state=42) param_grid = {'n_estimators': [100, 500, 1000, 2000, 3000, 4000, 5000], 'max_depth': [None, 5, 10, 15, 20], 'min_samples_split': [2, 5, 10, 20, 50], 'min_samples_leaf': [1, 5, 10, 20, 50], 'class_weight': ['balanced', None]} grid_search = GridSearchCV(rf, param_grid, cv=5, scoring='accuracy') grid_search.fit(X_train, y_train)```	```Best Random Forest Accuracy: 0.7577272727272727```

GRADIENT BOOSTING	```from sklearn.ensemble import GradientBoostingClassifier
	from sklearn.model_selection import GridSearchCV
	grid_search = GridSearchCV(GradientBoostingClassifier(
	random_state=42),{
	'learning_rate': [0.01, 0.1, 1],
	'n_estimators': [100, 200, 300],
	'max_depth': [3, 5, 7]},
	cv=5, scoring='accuracy').fit(X_train, y_train)```
		```Best Gradient Accuracy: 0.7667272727272727```

KNN	```## Tuned Hypermeter from sklearn.model_selection import GridSearchCV param_grid = {'n_neighbors': [3, 5, 7, 9, 11], 'weights': ['uniform', 'distance'], 'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute']} grid_search = GridSearchCV(KNeighborsClassifier(), param_grid, cv=5, scoring='accuracy') grid_search.fit(X_train, y_train)```	```Best KNN Accuracy: 0.7277272727272728 Best KNN Classification Report:```

Logistic Regression		

Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric
Random Forest		
GRADIENT BOOSTING		

KNN	KNN Classification Report: precision recall f1-score support 0 0.57 0.59 0.58 720 1 0.80 0.78 0.79 1480 accuracy 0.72 2200 macro avg 0.68 0.69 0.68 2200 weighted avg 0.72 0.72 0.72 2200	Best KNN Accuracy: 0.7277272727272728 Best KNN Classification Report: precision recall f1-score support 0 0.58 0.64 0.61 720 1 0.82 0.77 0.79 1480 accuracy 0.73 2200 macro avg 0.70 0.71 0.70 2200 weighted avg 0.74 0.73 0.73 2200 Best KNN Confusion Matrix: [[462 258] [341 1139]]
Logistic Regression	Logistic Regression Classification Report: precision recall f1-score support 0 0.46 0.25 0.32 720 1 0.70 0.85 0.77 1480 accuracy 0.66 2200 macro avg 0.58 0.55 0.55 2200 weighted avg 0.62 0.66 0.62 2200	Best Logistic Regression Accuracy: 0.6631818181818182 Best Logistic Regression Classification Report: precision recall f1-score support 0 0.61 0.60 0.61 720 1 0.81 0.81 0.81 1480 accuracy 0.74 2200 macro avg 0.71 0.71 0.71 2200 weighted avg 0.74 0.74 0.74 2200 Best Logistic Regression Confusion Matrix: [[434 286] [279 1201]]

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.