Model Optimization and Tuning Phase Template

Date	18 July 2024
Team ID	SWTID1720190389
Project Title	E-Commerce Shipping Prediction
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

Logistic Regression	`param_grids = {`
	
	`'LogisticRegression': {`
	`'solver': ['liblinear', 'lbfgs'],`
	`'C': [0.01, 0.1, 1, 10, 100]`
	`},`
		ACC--> 0.6803092314688495
		R2CV--> 0.6884973017849729
		MEAN SQUARED ERROR--> 0.5581242677173489
		ROC--> (array([0. , 0.30954995, 1.]), array([0.
		------------------------------ 	

GaussianNB	`param_grids = {`
	
	`'GaussianNB': {`
	`'var_smoothing': [1e-09, 1e-08, 1e-07, 1e-06, 1e-05]`
	`},`
		ACC--> 0.6821282401091405
		R2CV--> 0.6712224989622251
		MEAN SQUARED ERROR--> 0.5733912286020558
		ROC--> (array([0. , 0.13391877, 1.]), array([0.
		------------------------------ 	

KNN	`param_grids = {`	ACC--> 0.6630286493860846 R2CV--> 0.6652988792029887 MEAN SQUARED ERROR--> 0.5785335952189908 ROC--> (array([0. , 0.37211855, 1.]), arr ------------------------------

	'KNeighborsClassifier': { 'n_neighbors': [3, 5, 7, 9, 11], 'weights': ['uniform', 'distance'] },	

Decision Tree	param_grids = {
	
	'DecisionTreeClassifier': {
	'max_depth': [3, 5, 7, 9, None],
	'min_samples_split': [2, 5, 10],
	'min_samples_leaf': [1, 2, 4]
	},
		ACC--> 0.6912232833105957
		R2CV--> 0.6898713158987132
		MEAN SQUARED ERROR--> 0.556891986027171
		ROC--> (array([0. , 0.09110867, 1.]), array([0
		------------------------------ 	

Random Forest	param_grids = {
	
	'RandomForestClassifier': {
	'n_estimators': [50, 100, 200],
	'max_depth': [3, 5, 7, 9, None],
	'min_samples_split': [2, 5, 10],
	'min_samples_leaf': [1, 2, 4]
	},
		ACC--> 0.6966803092314688
		R2CV--> 0.6816791199667911
		MEAN SQUARED ERROR--> 0.5641993265089996
		ROC--> (array([0. , 0.09110867, 1.]), array
		------------------------------ 	

Gradient Boosting	param_grids = {
	
	'GradientBoostingClassifier': {
	'n_estimators': [50, 100, 200],
	'learning_rate': [0.01, 0.1, 0.05],
	'max_depth': [3, 5, 7, 9]
	},
		ACC--> 0.694406548431105
		R2CV--> 0.6775778331257782
		MEAN SQUARED ERROR--> 0.5678223021986911
		ROC--> (array([0. , 0.09659715, 1.]), arra
		------------------------------ 	

XGBoost	param_grids = {
	
	'XGBClassifier': {
	'n_estimators': [50, 100, 200],
	'learning_rate': [0.01, 0.1, 0.05],
	'max_depth': [3, 5, 7, 9]
	},
		ACC--> 0.700318326512051
		R2CV--> 0.6934952262349523
		MEAN SQUARED ERROR--> 0.5536287327849302
		ROC--> (array([0. , 0.07244786, 1.]), array
		------------------------------ 	

CatBoost	param_grids = {	ACC--> 0.6998635743519782 R2CV--> 0.6885014528850146 MEAN SQUARED ERROR--> 0.5581205489094497 ROC--> (array([0. , 0.08122942, 1.]), array([------------------------------

	'CatBoostClassifier': {
	'iterations': [50, 100, 200],
	'learning_rate': [0.01, 0.1, 0.05],
	'depth': [3, 5, 7, 9]
	}	

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric
Logistic Regression	LogisticRegression: ---------- classification_report -->

	precision	recall	f1-score	support
0	0.60	0.69	0.64	911
1	0.75	0.67	0.71	1288
accuracy			0.68	2199
macro avg	0.68	0.68	0.68	2199
weighted avg	0.69	0.68	0.68	2199

confusion_matrix -->
[[629 282]
[421 867]] |
| GaussianNB | ----------
GaussianNB:
----
classification_report -->

	precision	recall	f1-score	support
0	0.58	0.87	0.69	911
1	0.85	0.55	0.67	1288
accuracy			0.68	2199
macro avg	0.72	0.71	0.68	2199
weighted avg	0.74	0.68	0.68	2199

confusion_matrix -->
[[789 122]
[577 711]] |

KNN	KNeighborsClassifier: ---------- classification_report -->

	precision	recall	f1-score	support
0	0.59	0.63	0.61	911
1	0.72	0.69	0.71	1288
accuracy			0.66	2199
macro avg	0.66	0.66	0.66	2199
weighted avg	0.67	0.66	0.66	2199

confusion_matrix -->
[[572 339]
 [402 886]] |
| Decision Tree | DecisionTreeClassifier:
----
classification_report -->

	precision	recall	f1-score	support
0	0.58	0.91	0.71	911
1	0.89	0.54	0.67	1288
accuracy			0.69	2199
macro avg	0.74	0.72	0.69	2199
weighted avg	0.76	0.69	0.69	2199

confusion_matrix -->
[[828 83]
 [596 692]] |

Random Forest	RandomForestClassifier: ---------- classification_report -->

	precision	recall	f1-score	support
0	0.59	0.91	0.71	911
1	0.89	0.55	0.68	1288
accuracy			0.70	2199
macro avg	0.74	0.73	0.70	2199
weighted avg	0.77	0.70	0.69	2199

confusion_matrix -->
[[828 83]
[584 704]] |
| Gradient Boosting | GradientBoostingClassifier:
----
classification_report -->

	precision	recall	f1-score	support
0	0.58	0.90	0.71	911
1	0.89	0.55	0.68	1288
accuracy			0.69	2199
macro avg	0.74	0.72	0.69	2199
weighted avg	0.76	0.69	0.69	2199

confusion_matrix -->
[[823 88]
[584 704]] |

XGBoost	XGBClassifier: ---------- classification_report --> precision recall f1-score support 0 0.59 0.93 0.72 911 1 0.91 0.54 0.68 1288 accuracy 0.70 2199 macro avg 0.75 0.73 0.70 2199 weighted avg 0.78 0.70 0.70 2199 confusion_matrix --> [[845 66] [593 695]]
CatBoost	CatBoostClassifier: ---------- classification_report --> precision recall f1-score support 0 0.59 0.92 0.72 911 1 0.90 0.55 0.68 1288 accuracy 0.70 2199 macro avg 0.75 0.73 0.70 2199 weighted avg 0.77 0.70 0.70 2199 confusion_matrix --> [[837 74] [586 702]]

Model Evaluation after Hyper Parameter Tuning

	Name	Accuracy	f1_score	Recall	Precision
0	LogisticRegression	68.03	71.15	67.31	75.46
1	GaussianNB	68.21	67.04	55.20	85.35
2	KNeighborsClassifier	66.30	70.51	68.79	72.33
3	DecisionTreeClassifier	69.12	67.09	53.73	89.29
4	RandomForestClassifier	69.67	67.86	54.66	89.45
5	GradientBoostingClassifier	69.44	67.69	54.66	88.89
6	XGBClassifier	70.03	67.84	53.96	91.33
7	CatBoostClassifier	69.99	68.02	54.50	90.46

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
XGBoost	The XGBoost 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