Model Optimization and Tuning Phase Template

Date	10 July 2024
Team ID	SWTID1720097765
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
Logistic Regression	`lg = LogisticRegressionCV(n_jobs=-1,random_state= 1234)` `lg_param_grid = {` `'Cs': [6,8,10,15,20],` `'max_iter': [60,80,100]` `}` `lg_cv= GridSearchCV(lg,lg_param_grid,cv=5,scoring="accuracy", n_jobs=-1, verbose=3)` `lg_cv.fit(x_train_normalized,y_train)` `print("Best Score:" + str(lg_cv.best_score_))` `print("Best Parameters: " + str(lg_cv.best_params_))`	`Fitting 5 folds for each of 15 candidates, totalling 75 fits` `Best Score:0.6356404077730116` `Best Parameters: {'Cs': 6, 'max_iter': 60}`
SVM	`svc = svm.SVC(random_state=1234,kernel='rbf',C= 10, gamma= 5 , tol = 1e-2,verbose = 1)` `svc.fit(x_train_normalized, y_train)` `print('train score',svc.score(x_train_normalized,y_train))` `print('test score',svc.score(x_test_normalized,y_test))`	`[LibSVM]train score 0.6650755767700876` `test score 0.6668181818181819`

XG Boost	``` params = { 'min_child_weight': [10,20], 'gamma': [1.5, 2.0, 2.5], 'colsample_bytree': [0.6, 0.8, 0.9], 'max_depth': [4,5,6] } xgb = XGBClassifier(learning_rate=0.5, n_estimators=100, objective='binary:logistic', nthread=3) fitmodel = GridSearchCV(xgb, param_grid=params, cv=5, refit=True, scoring="accuracy", n_jobs=-1, verbose=3) fitmodel.fit(x_train_normalized, y_train) print(fitmodel.best_estimator_, fitmodel.best_params_, fitmodel.best_score_) ```	
Random Forest	``` rf = RandomForestClassifier() rf_param_grid = { 'n_estimators': [200,300,500], 'criterion': ['entropy','gini'], 'max_depth': [7,8,60,80,100], 'max_features': ['auto', 'sqrt', 'log2'] } rf_cv= GridSearchCV(rf,rf_param_grid,cv=7,scoring="accuracy", n_jobs=-1, verbose=3) rf_cv.fit(x_train,y_train) print("Best Score:" + str(rf_cv.best_score_)) print("Best Parameters: " + str(rf_cv.best_params_)) ```	train 0.7104216388225935 test 0.6818181818181818

Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric
MODELS		

Baseline Metric:

	Name	Accuracy	f1_score	Recall	Precision
0	logistic regression	59.27	74.43	100.00	59.27
1	logistic regression CV	63.27	66.78	62.27	71.99
2	XGBoost	65.59	69.63	66.56	73.00
3	Ridge classifier	59.27	74.43	100.00	59.27
4	KNN	63.27	68.39	67.02	69.81
5	Random Forest	67.18	70.36	65.72	75.71
6	Support Vector Classifier	59.27	74.43	100.00	59.27

Optimized Metric:

	Name	Accuracy	f1_score	Recall	Precision
0	logistic regression	59.27	74.43	100.00	59.27
1	logistic regression CV	63.27	66.78	62.27	71.99
2	XGBoost	65.27	68.84	64.72	73.52
3	Ridge classifier	59.27	74.43	100.00	59.27
4	KNN	63.27	68.39	67.02	69.81
5	Random Forest	67.55	66.38	54.06	85.98
6	Support Vector Classifier	67.05	69.06	62.04	77.86

Final Model Selection Justification (2 Marks):

Final Model	Reasoning

Random Forest	The Random Forest 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.