

Model Optimization and Tuning Phase Template

Date	15 March 2024
Team ID	SWTID1720000747
Project Title	Detection Of Autistic Spectrum Disorder: Classification
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
KNN	<pre>param_grid_knn = { 'n_neighbors': [3, 5, 7, 9], 'weights': ['uniform', 'distance'] } grid_search_knn = GridSearchCV(estimator=KNeighborsClassifier(), param_grid=param_grid_knn, cv=5, scoring='accuracy') grid_search_knn.fit(X_train, y_train) best_knn = grid_search_knn.best_estimator_</pre>	<pre>print(f'Best parameters for KNN: {grid_search_knn.best_params_}') print(f'Accuracy for KNN: {accuracy_score(y_test, best_knn.predict(X_test)):.4f}')</pre> <p>Best parameters for KNN: {'n_neighbors': 7, 'weights': 'uniform'} Accuracy for KNN: 0.9672</p>
Logistic Regression	<pre># Hyper Parameter tuning param_grid_lr = { 'C': [0.1, 1, 10, 100], 'solver': ['liblinear', 'lbfgs'] } grid_search_lr = GridSearchCV(estimator=LogisticRegression(), param_grid=param_grid_lr, cv=5, scoring='accuracy') grid_search_lr.fit(X_train, y_train) best_lr = grid_search_lr.best_estimator_</pre>	<pre>print(f'Best parameters for Logistic Regression: {grid_search_knn.best_params_}') print(f'Accuracy for Logistic Regression: {accuracy_score(y_test, best_knn.predict(X_test)):.4f}')</pre> <p>Best parameters for Logistic Regression: {'max_depth': None, 'min_samples_split': 2, 'n_estimators': 100} Accuracy for Logistic Regression: 0.972</p>

SVM	<pre>param_grid_svm = { 'C': [0.1, 1, 10, 100], 'kernel': ['linear', 'rbf'] } grid_search_svm = GridSearchCV(estimator=SVC(), param_grid=param_grid_svm, cv=5, scoring='accuracy') grid_search_svm.fit(X_train, y_train) best_svm = grid_search_svm.best_estimator_</pre>	<pre>print(f'Best parameters for SVM: {grid_search_svm.best_params_}') print(f'Accuracy for SVM: {accuracy_score(y_test, best_svm.predict(X_test)):.4f}')</pre> <p>Best parameters for SVM: {'C': 0.1, 'kernel': 'linear'} Accuracy for SVM: 1.0000</p>
Decision Trees	<pre>param_grid_dt = { 'max_depth': [None, 10, 20, 30], 'min_samples_split': [2, 5, 10] } grid_search_dt = GridSearchCV(estimator=DecisionTreeClassifier(), param_grid=param_grid_dt, cv=5, scoring='accuracy') grid_search_dt.fit(X_train, y_train) best_dt = grid_search_dt.best_estimator_</pre>	<pre>print(f'Best parameters for Decision Tree: {grid_search_dt.best_params_}') print(f'Accuracy for Decision Tree: {accuracy_score(y_test, best_dt.predict(X_test)):.4f}')</pre> <p>Best parameters for Decision Tree: {'max_depth': None, 'min_samples_split': 2} Accuracy for Decision Tree: 1.0000</p>
Random Forest	<pre>param_grid_rf = { 'n_estimators': [100, 200, 300], 'max_depth': [None, 10, 20], 'min_samples_split': [2, 5, 10] } grid_search_rf = GridSearchCV(estimator=RandomForestClassifier(), param_grid=param_grid_rf, cv=5, scoring='accuracy') grid_search_rf.fit(X_train, y_train) best_rf = grid_search_rf.best_estimator_</pre>	<pre>print(f'Best parameters for Random Forest: {grid_search_rf.best_params_}') print(f'Accuracy for Random Forest: {accuracy_score(y_test, best_rf.predict(X_test)):.4f}')</pre> <p>Best parameters for Random Forest: {'max_depth': None, 'min_samples_split': 2, Accuracy for Random Forest: 1.0000</p>

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric				
KNN	precision	recall	f1-score	support	
	0	0.99	0.97	0.98	87
	1	0.92	0.97	0.94	35
	accuracy			0.97	122
	macro avg	0.95	0.97	0.96	122
	weighted avg	0.97	0.97	0.97	122
Logistic Regression	precision	recall	f1-score	support	
	0	0.98	1.00	0.99	41
	1	1.00	0.95	0.97	20
	accuracy			0.98	61
	macro avg	0.99	0.97	0.98	61
	weighted avg	0.98	0.98	0.98	61

SVM		precision	recall	f1-score	support
	0	0.97	1.00	0.98	87
	1	1.00	0.91	0.96	35
	accuracy			0.98	122
	macro avg	0.98	0.96	0.97	122
	weighted avg	0.98	0.98	0.98	122
Decision Trees		precision	recall	f1-score	support
	0	0.98	0.98	0.98	41
	1	0.95	0.95	0.95	20
	accuracy			0.97	61
	macro avg	0.96	0.96	0.96	61
	weighted avg	0.97	0.97	0.97	61
Random Forest		precision	recall	f1-score	support
	0	0.98	0.98	0.98	41
	1	0.95	0.95	0.95	20
	accuracy			0.97	61
	macro avg	0.96	0.96	0.96	61
	weighted avg	0.97	0.97	0.97	61

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest	<p>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.</p>