Model Optimization and Tuning Phase Template

Date	16 July 2024
Team ID	739870
Project Title	Freedom Of The World Classification
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining neural network 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 (8 Marks):

Model	Tuned Hyperparameters
KNN	The provided code demonstrates hyperparameter tuning for a K-Nearest neighbores(KNN) classifier using gridsearchCV. It defines a parameter grid (kn_param_grid) with different values for the number of neighbors (n_neighbors), the weight function (weight),and the algorithm used to computer the nearest neighbors (algorithm).gridsearchCV is configured with 5-fold cross-validation (cv=5) and evaluates model performance based on accuracy (scoring='accuracy'). The best hyperparameters and the best KNN model are determined bt fitting the model to the training data.

```
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier

# Define the parameter grid
knn_param_grid = {
    'n_neighbors': [3, 5, 7, 9, 11, 13],
    'weights': ['uniform', 'distance'],
    'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute']
}

# Initialize the KNN classifier
knn = KNeighborsClassifier()

# Initialize GridSearchCV
grid_search = GridSearchCV(estimator=knn, param_grid=knn_param_grid, cv=5, scoring='accuracy')

# Fit the model
# Use x_train instead of X_train
grid_search.fit(x_train, y_train)

# Get the best parameters
best_params = grid_search.best_params_
print(f"Best parameters: {best_params}")

# Get the best estimator
best_knn = grid_search.best_estimator_
print(f"Best KNN model: {best_knn}")
```

SVM	The code demonstrates hyperparameter tuning for a support vector machine (SVM) classifier using GridsearchCV. It define aparameter grid (svm_param_grid) with various values for the regukarization parameter(C), Kernel type (kernel),and kernel coefficient (gamma).GridsearchCV is configured with 5-fold cross-validation (cv=5) and evaluates model performance based on accuracy (scoring='accuracy'). The best hyperparameters and the best SVM model are determined by fitting the moedl to the training data (x_train,y_train).
	`from sklearn.model_selection import GridSearchCV`
	`from sklearn.svm import SVC`
	
	`# Define the parameter grid for SVM`
	`svm_param_grid = {`
	`'C': [0.1, 1, 10, 100],`
	`'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],`
	`'gamma': ['scale', 'auto']`
	`}`
	
	`# Initialize the SVM classifier`
	`svm = SVC()`
	
	`# Initialize GridSearchCV`
	`grid_search_svm = GridSearchCV(estimator=svm, param_grid=svm_param_grid, cv=5, scoring='accuracy')`
	
	`# Fit the model`
	`grid_search_svm.fit(x_train, y_train)`
	
	`# Get the best parameters`
	`best_params_svm = grid_search_svm.best_params_`
	`print(f"Best parameters: {best_params_svm}")`
	
	`# Get the best estimator`
	`best_svm = grid_search_svm.best_estimator_`
	`print(f"Best SVM model: {best_svm}")`

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
KNN	K-Nearest Neighbors (KNN) is chosen for its simplicity and effectiveness in classification tasks, particularly in scenarios where the relationship between features is non-linear. KNN is advantageous due to its intuitive nature and its ability to handle multi-class classification problems without significant modifications. Above two model, KNN model have the highest accuracy among the models.