

Model Optimization and Tuning Phase Template

Date	02 November 2024
Team ID	SWTID1726834817
Project Title	Fake News Analysis in social media in NLP.
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
Gradient Boosting	<pre># Define the parameter grid for Gradient Boosting gb_param_grid = { 'n_estimators': [50, 100, 200], 'learning_rate': [0.01, 0.1, 0.2], 'max_depth': [3, 4, 5], 'min_samples_split': [2, 5, 10] } # Initialize the Gradient Boosting classifier gb_classifier = GradientBoostingClassifier(random_state=53)</pre>	<pre># Calculate the accuracy score gb_score = accuracy_score(y_test, gb_pred) print("Gradient Boosting Model Accuracy:", gb_score)</pre> <p>Gradient Boosting Model Accuracy: 0.8939177427868389</p>
Decision Tree	<pre># Define the hyperparameters and their ranges to try param_grid = { 'max_depth': [10, 20, 30, None], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4] } # Initialize the Decision Tree Classifier dt_classifier = DecisionTreeClassifier(random_state=53)</pre>	<pre>dt_score = accuracy_score(y_test, dt_pred) print("Decision Tree Model Accuracy:", dt_score)</pre> <pre># Display the confusion matrix dt_cm = confusion_matrix(y_test, dt_pred, labels=['FAKE', 'REAL']) print("Confusion Matrix:\n", dt_cm)</pre> <p>Decision Tree Model Accuracy: 0.838616929687888</p>
Random Forest	<pre># Define the parameter grid for Random Forest rf_param_grid = { 'n_estimators': [50, 100, 200], 'max_depth': [None, 10, 20, 30], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4] } # Initialize the Random Forest classifier rf_classifier = RandomForestClassifier(random_state=53)</pre>	<pre># Calculate the accuracy score rf_score = accuracy_score(y_test, rf_pred) print("Random Forest Model Accuracy:", rf_score)</pre> <p>Random Forest Model Accuracy: 0.9876996652319465</p>

KNN	<pre>knn_param_grid = { 'n_neighbors': [3, 5, 7, 9, 11], 'weights': ['uniform', 'distance'], 'metric': ['euclidean', 'manhattan'] } # Initialize the KNN classifier knn_classifier = KNeighborsClassifier()</pre>	<pre>knn_score = accuracy_score(y_test, knn_pred) print("KNN Model Accuracy:", knn_score)</pre> <p>Best parameters found for KNN: {'metric': 'euclidean', 'n_neighbors': 5, 'weights': 'distance'} Tuned KNN Model accuracy: 0.574565233866333</p>
-----	--	--

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric
Gradient Boosting	<pre># Display the confusion matrix gb_cm = confusion_matrix(y_test, gb_pred, labels=['FAKE', 'REAL']) print("Confusion Matrix:\n", gb_cm)</pre> <p>Confusion Matrix: [[927 81] [145 938]]</p>
Decision Tree	<pre># Display the confusion matrix dt_cm = confusion_matrix(y_test, dt_pred, labels=['FAKE', 'REAL']) print("Confusion Matrix:\n", dt_cm)</pre> <p>Decision Tree Model Accuracy: 0.8386149224987888 Confusion Matrix: [[838 198] [198 885]]</p>
Random Forest	<pre># Display the confusion matrix rf_cm = confusion_matrix(y_test, rf_pred, labels=['FAKE', 'REAL']) print("Confusion Matrix:\n", rf_cm)</pre> <p>Confusion Matrix: [[931 77] [116 967]]</p>

KNN	<pre># Display the confusion matrix knn_cm = confusion_matrix(y_test, knn_pred, labels=['FAKE', 'REAL']) print("Confusion Matrix:\n", knn_cm)</pre> <p>Confusion Matrix: [[1006 2] [951 132]]</p>
-----	---

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Naïve Bayes	Naïve Bayes is used in the provided code because it is a common and effective baseline model for text classification tasks, especially with high-dimensional datasets like text data. It is included because of its efficiency, suitability for high-dimensional text data, effectiveness in binary classification, and role as a baseline for comparing other algorithms.