



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	# Define the parameter grid for Gradient Boosting gb_param_grid = { 'n_estimators': [50, 100, 200], 'learning_rate': [0.01, 0.1, 0.2], 'mex_depth': [3, 4, 5], 'min_samples_split': [2, 5, 10] } # Initialize the Gradient Boosting classifier gb_classifier = GradientBoostingClassifier(random_state=53)	# calculate the accuracy scare gb_score = accuracy_score(y_test, gb_pred) print("Gradient Boosting Model Accuracy:", gb_score) Gradient Boosting Model Accuracy: 0.8919177427068389
Decision Tree	# Define the hyperporometers and their ranges to try param_grid = { 'max_depth': [10, 20, 30, Name], 'min_samples_plit': [2, 5, 30], 'min_samples_leaf': [1, 2, 4] } # Initialize the Decision Tree Classifier dt_classifier = DecisionTreeClassifier(random_state=53)	dt_score = accuracy_score(y_test, dt_pred) print("Decision Tree rodel Accuracy;", dt_score) # Display the confusion motrix dt_cm = confusion_metrix(y_test, dt_pred, labels=["FAME", 'REAL"]) print("Confusion Petrix(\n", dt_cm) Decision Tree rodel Accuracy; 0.8306169296987888
Random Forest	<pre>rf_param_grid = { 'n_estimators': [50, 300, 200], 'mat_depth': [Name, 30, 30, 30], 'min_samples_split': [2, 5, 30], 'min_samples_split': [2, 5, 30], 'min_samples_ter': [1, 2, 4] } # Initialize the Mandow Forest classifier rf_classifier = RandomForestclassifier(random_state=53)</pre>	# Colculate the occuracy scare rf_score = accuracy_score(y_test, rf_pred) print("Random Forest Model Accuracy:", rf_score) Random Forest Model Accuracy: 0.9076996552319465



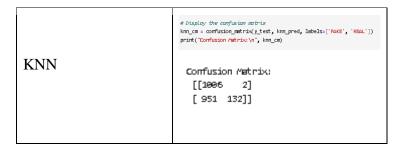


Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric
Gradient Boosting	# Display the confusion matrix gb_cm = confusion_matrix(p_test, gb_pred, labels=['FAKE', 'REAL']) print('Confusion retrix(\n', gb_cm) Confusion Matrix: [[927 81] [145 938]]
Decision Tree	# Display the confusion matrix dt_cm = confusion_matrix(y_test, dt_pred, labels=['FAKE', 'REAL']) print('Confusion Matrix: dt_cm' Decision Tree Model Accuracy: 0.8386169296987888 Confusion Matrix: [[398 888]]
Random Forest	# Display the confusion matrix rf_cm = confusion matrix(y_test, rf_pred, labels=['FAKE', 'RBAL']) print("Confusion Metrix:\n", rf_cm) Confusion Metrix: [[931 77] [116 967]]







Final Model Selection Justification (2 Marks):

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