Model Optimization and Tuning Phase Template

Date	Nov 30, 2024
Team ID	739891
Project Title	Unlocking the Minds: Analyzing Mental Health with NLP
Maximum Marks	10 Marks

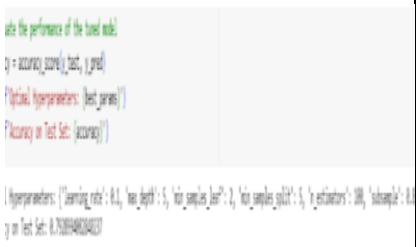
Model Optimization and Tuning Phase

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
SVC	[25]: ```from sklearn.svm import SVC from sklearn.metrics import classification_report, accuracy_score sv = SVC() sv.fit(x_train,y_train) y_pred = sv.predict(x_test) print(classification_report(y_test,y_pred))```	
Decision tree classifier	[27]: ```from sklearn.tree import DecisionTreeClassifier Dt = DecisionTreeClassifier() Dt.fit(x_train,y_train) y_pred = Dt.predict(x_test) print(classification_report(y_test,y_pred)) print(accuracy_score(y_test,y_pred))```	

Random forest classifier	`from sklearn.ensemble import RandomForestClassifier`
	`rf = RandomForestClassifier()`
	`rf.fit(x_train,y_train)`
	`y_pred = rf.predict(x_test)`
	`print(classification_report(y_test,y_pred))`
	`print(accuracy_score(y_test,y_pred))`
		

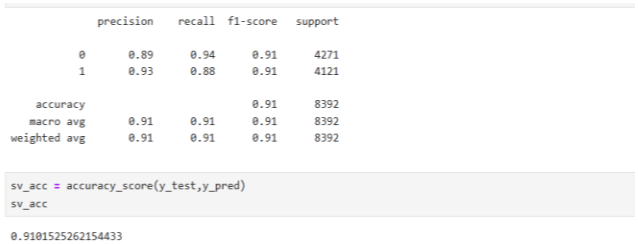
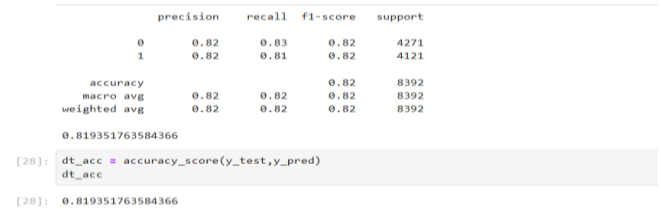
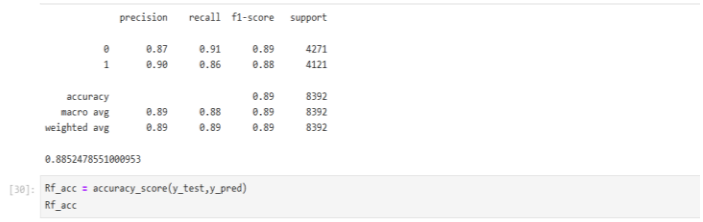
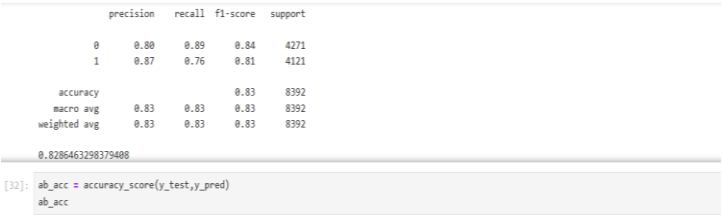
Ada boost classifier	`[31]: from sklearn.ensemble import AdaBoostClassifier`
	`# clf = AdaBoostClassifier(algorithm='SAMME')`
	`ab = AdaBoostClassifier(algorithm='SAMME')`
	`ab.fit(x_train,y_train)`
	`y_pred = ab.predict(x_test)`
	`print(classification_report(y_test,y_pred))`
	`print(accuracy_score(y_test,y_pred))`
		

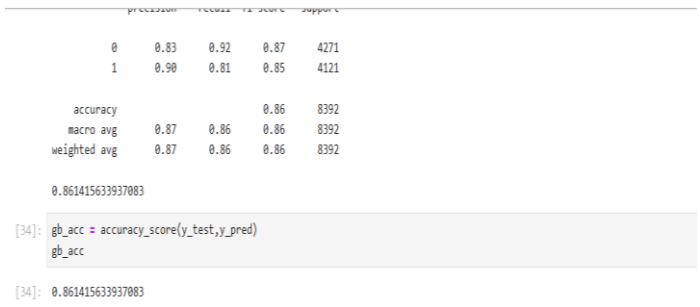
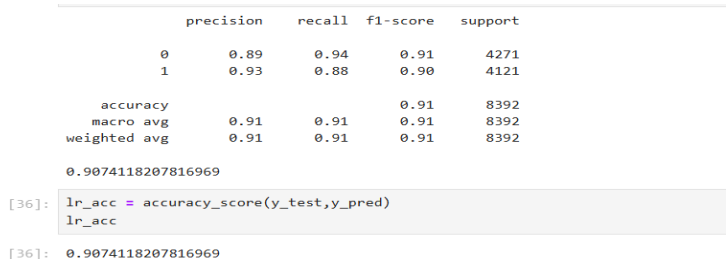
Gradient boosting classifier	`[3]: from sklearn.ensemble import GradientBoostingClassifier`
	`gb = GradientBoostingClassifier()`
	`gb.fit(x_train,y_train)`
	`y_pred = gb.predict(x_test)`
	`print(classification_report(y_test,y_pred))`
	`print(accuracy_score(y_test,y_pred))`
		

Logistic Regression	`[35]: from sklearn.linear_model import LogisticRegression`
	`lr = LogisticRegression()`
	`lr.fit(x_train,y_train)`
	`y_pred = lr.predict(x_test)`
	`print(classification_report(y_test,y_pred))`
	`print(accuracy_score(y_test,y_pred))`
		

Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric

SVC	
Decision tree classifier	
Random forest classifier	
Ada boost classifier	

Gradient boosting classifier	
Logistic Regression	

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
SVC (support vector classifier)	SVC performs well with limited data since it focuses on support vectors rather than the entire dataset. SVC's combination of flexibility (kernels), accuracy, robustness, and theoretical rigor makes it the "best" choice for projects where these qualities are critical and evaluating alternatives based on dataset's size, structure, and requirements is essential.