

Date	20 June 2024
Team ID	739903
Project Title	Mental health prediction
Maximum Marks	10 Marks

## Model Optimization and Tuning Phase Report

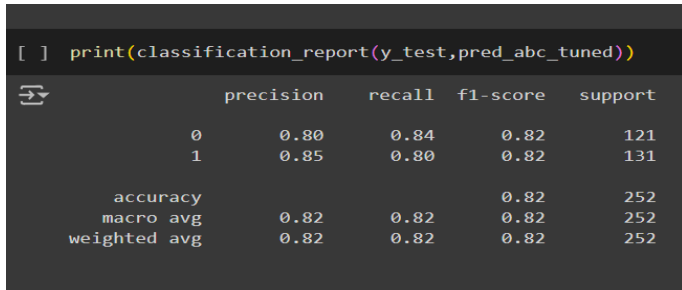
### Model Optimization and Tuning Phase:

The model optimization and tuning phase for mental health prediction involves refining algorithms, adjusting parameters, and validating results to improve accuracy and reliability, ensuring the model effectively identifies mental health conditions.

### Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
Random Forest	<pre> from sklearn.metrics import mean_squared_error params = {} n_estimators = 100 for n_estimators in range(10, 100, 10):     for learning_rate in [0.01, 0.05, 0.1, 0.5, 1.0]:         abc_random = RandomForestRegressor(n_estimators=n_estimators, learning_rate=learning_rate, max_depth=5, min_samples_split=10, min_samples_leaf=5)         abc_random.fit(X_train, y_train)         y_pred = abc_random.predict(X_test)         mse = mean_squared_error(y_test, y_pred)         params[n_estimators, learning_rate] = mse  # Find the best parameters best_params = min(params.items(), key=lambda item: item[1]) n_estimators, learning_rate = best_params  # Create the final model abc_random_best = RandomForestRegressor(n_estimators=n_estimators, learning_rate=learning_rate, max_depth=5, min_samples_split=10, min_samples_leaf=5) abc_random_best.fit(X_train, y_train) y_pred_final = abc_random_best.predict(X_test) mse_final = mean_squared_error(y_test, y_pred_final) </pre>	<pre> {'n_estimators': [1, 4, 8, 11, 15, 18, 22, 25, 29, 32, 36, 39, 43, 46, 50],  'learning_rate': [0.97, 0.98, 0.99, 1.0, 1.01, 1.02, 1.03, 1.04]} </pre>
AdaBoost Classifier	<pre> abc_tuned = AdaBoostClassifier(random_state=42, n_estimators=11, learning_rate=1.02) abc_tuned.fit(X_train_imputed, y_train) pred_abc_tuned = abc_tuned.predict(X_test_imputed) print('Accuracy of AdaBoost(tuned) =', accuracy_score(y_test, pred_abc_tuned)) </pre> <p>Accuracy of AdaBoost(tuned) = 0.8214285714285714</p>	<p>Accuracy of AdaBoost(tuned) = 0.8214285714285714</p>

### Performance Metrics Comparison Report (2 Marks):

Model	Optimized Metric																														
abc_tuned	<div><pre>[ ] print(classification_report(y_test,pred_abc_tuned))</pre><table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.80</td><td>0.84</td><td>0.82</td><td>121</td></tr><tr><td>1</td><td>0.85</td><td>0.80</td><td>0.82</td><td>131</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.82</td><td>252</td></tr><tr><td>macro avg</td><td>0.82</td><td>0.82</td><td>0.82</td><td>252</td></tr><tr><td>weighted avg</td><td>0.82</td><td>0.82</td><td>0.82</td><td>252</td></tr></tbody></table></div>		precision	recall	f1-score	support	0	0.80	0.84	0.82	121	1	0.85	0.80	0.82	131	accuracy			0.82	252	macro avg	0.82	0.82	0.82	252	weighted avg	0.82	0.82	0.82	252
	precision	recall	f1-score	support																											
0	0.80	0.84	0.82	121																											
1	0.85	0.80	0.82	131																											
accuracy			0.82	252																											
macro avg	0.82	0.82	0.82	252																											
weighted avg	0.82	0.82	0.82	252																											

### Final Model Selection Justification (2 Marks):

Final Model	Reasoning
XGB Clasiifier	The XGB Classifier 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.