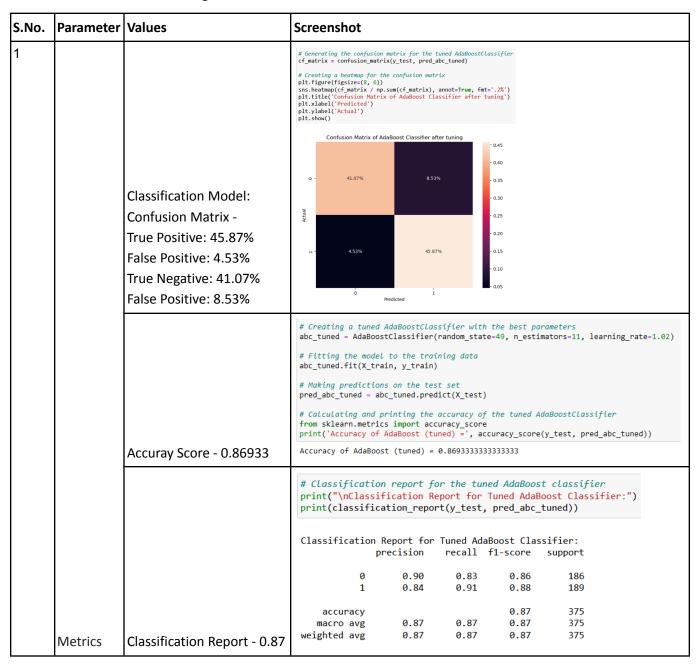
Project Development Phase V Model Performance Test

Date	18 November 2022
Team ID	591761
Project Name	Predicting Mental Health Illness Of Working
	Professionals Using Machine Learning
Maximum Marks	10 Marks

Model Performance Testing:



2			import unity as np free silearn accel_calcetion import RandomizedSearch() parast_abc - { par
			pares, de: -(" [.stsfarers': [in(s) for x in ro,limpsec(starts_1.stm-58, num-15)], [.stsfarers': [(x,0) = x / 280) for x in rosp(6, 0)]]
			# Assembly label is your condent Boosting Classifier # Replace "estent-on-sel" left) your actual Gradient Boosting Classifier Instance abc_rendom = Remonstratementh(VI) rendom, tracts re
			estimator-abc, parama_distribution-paramas_abc,
			n_idos1 }
			parass_abc ("n_stitators": [1, 4, 8, 11, 15, 18, 22, 25, 20, 32, 36, 30, 43, 46, 58], 'learning_rest': [0.97, 0.98, 0.99, 1.0, 1.00, 1.02, 1.03, 1.04])
			abc_rendom.fit(X_train,y_train)
			RendomizedSearchCV(co.5, estatant-addisonctCastfor(rendom_tsta-600),
			1.05; 1.00; 1.105 'n_estimators': [1, 4, 9, 11, 15, 18, 20, 25, 29, 12, 10, 10, 10,
		Hyperparameter Tuning -	random_state=46) 42, 40, 30]], abc_random_sbat_params_
		0.8693	('n_estimators': 11, 'learning_rate': 1.02)
		(We can see an	# Creating a tuned Additional Classifier with the best parameters dectumed - Additional Classifier (reading atto eds. n. estimators-11, learning_rate-1.02) # FICTING the model to the troising data eds. n. estimators of the control of the contro
		improvement of 0.5% from	abc_tunedfift(Lftrain, x_train) # Number predictions on the test seet pred_ubc_tuned = abc_tunedfredict(Lftett)
		·	a Calculating and printing the accuracy of the tuned Adabascillosifler from silema.metrics import accuracy_score printi/accuracy_acfaboox (tune) -, accuracy_score(y_test, pred_abc_tuned))
		non boosted model)	Accuracy of Adaboust (tuned) = 0.8693333333333
		Validation Method -	from sklearn.metrics import accuracy_score
		Train-Test Split:	<pre>def model_test(X_train, X_test, y_train, y_test, model, model_name): model.fit(X_train, y_train)</pre>
		Train rese spire.	<pre>y_pred = model.predict(X_test) accuracy = accuracy_score(y_test, y_pred) print(f'Score for (model_name) is: (accuracy)')</pre>
		6	print()
		Score for Logistic	<pre># Loop through the models in the modeL_dict and test each one for model_name, model in model_dict.items(): model_test(X_train, X_test, y_train, y_test, model, model_name)</pre>
		regression is: 0.848	Score for Logistic regression is: 0.848
			Score for KNN Classifier is: 0.776
		Score for KNN Classifier is:	Score for Decision Tree Classifier is: 0.7946666666666666
		0.776	C:\Users\mp4vilanaconda3\iib\site-packages\sklearn\neighbors_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` will become false, the `axis` over which the statistic
		0.770	is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
			mode, _ = stats.mode(_y[neigh_ind, k], axis=1) Score for Random Forest Classifier is: 0.85333333333334
		Score for Decision Tree	Score for AdaBoost Classifier is: 0.864
		Classifier is:	Score for Gradient Boosting Classifier is: 0.84 Score for XGB Classifier is: 0.84
		0.794666666666666	36/10/ 7/00/ 0183321741/ 13. 0.04
		Score for Dandom Forest	
		Score for Random Forest	
		Classifier is:	
		0.853333333333334	
		Score for AdaBoost	
		Classifier is: 0.864	
		0.004	
		Score for Gradient	
		Boosting Classifier is: 0.84	
		Score for XGB Classifier is:	
	Tune the	0.84	
		0.04	
	Model		