Project Development Phase V
Model Performance Test

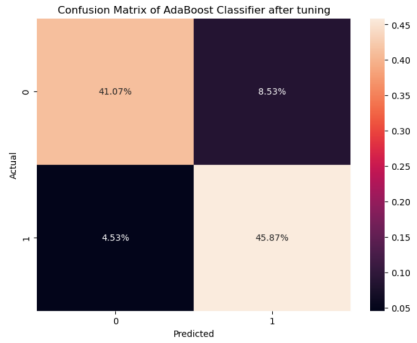
Date	18 November 2022
Team ID	591761
Project Name	Predicting Mental Health Illness Of Working Professionals Using Machine Learning
Maximum Marks	10 Marks

Model Performance Testing:

<table>
<tr><th>S.No.</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr><td rowspan="3">1</td><td rowspan="3">Metrics</td><td>Classification Model:
Confusion Matrix -
True Positive: 45.87%
False Positive: 4.53%
True Negative: 41.07%
False Positive: 8.53%</td><td>

```
# Generating the confusion matrix for the tuned AdaBoostClassifier
cf_matrix = confusion_matrix(y_test, pred_abc_tuned)

# Creating a heatmap for the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cf_matrix / np.sum(cf_matrix), annot=True, fmt='.2%')
plt.title('Confusion Matrix of AdaBoost Classifier after tuning')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```



</td></tr>
<tr><td>Accuray Score - 0.86933</td><td>

```
# Creating a tuned AdaBoostClassifier with the best parameters
abc_tuned = AdaBoostClassifier(random_state=49, n_estimators=11, learning_rate=1.02)

# Fitting the model to the training data
abc_tuned.fit(X_train, y_train)

# Making predictions on the test set
pred_abc_tuned = abc_tuned.predict(X_test)

# Calculating and printing the accuracy of the tuned AdaBoostClassifier
from sklearn.metrics import accuracy_score
print('Accuracy of AdaBoost (tuned) =', accuracy_score(y_test, pred_abc_tuned))
```

```
Accuracy of AdaBoost (tuned) = 0.8693333333333333
```

</td></tr>
<tr><td>Classification Report - 0.87</td><td>

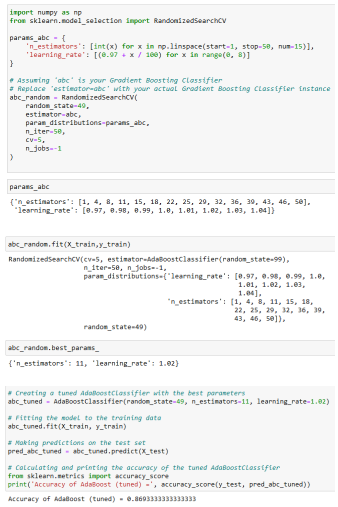
```
# Classification report for the tuned AdaBoost classifier
print("\nClassification Report for Tuned AdaBoost Classifier:")
print(classification_report(y_test, pred_abc_tuned))
```

```
Classification Report for Tuned AdaBoost Classifier:
              precision    recall  f1-score   support

           0       0.90      0.83      0.86       186
           1       0.84      0.91      0.88       189

    accuracy                           0.87       375
   macro avg       0.87      0.87      0.87       375
weighted avg       0.87      0.87      0.87       375
```

</td></tr>
</table>

2	Tune the Model	Hyperparameter Tuning - 0.8693 (We can see an improvement of 0.5% from non boosted model)	
		Validation Method - Train-Test Split: Score for Logistic regression is: 0.848 Score for KNN Classifier is: 0.776 Score for Decision Tree Classifier is: 0.7946666666666666 Score for Random Forest Classifier is: 0.8533333333333334 Score for AdaBoost Classifier is: 0.864 Score for Gradient Boosting Classifier is: 0.84 Score for XGB Classifier is: 0.84