Model Development Phase Template

Date	19 May 2025
Team ID	SWTID1750233055
Project Title	Mental Health Prediction
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

```python
#import model we need to train our Model
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from xgboost.sklearn import XGBClassifier
from sklearn.metrics import accuracy_score, roc_curve, confusion_matrix, classification_report, auc
```

```python
#Build the models
model_dict = {}

model_dict['Logistic regression']= LogisticRegression(solver='liblinear',random_state=49)
model_dict['KNN Classifier'] = KNeighborsClassifier()
model_dict['Decision Tree Classifier'] = DecisionTreeClassifier(random_state=49)
model_dict['Random Forest Classifier'] = RandomForestClassifier(random_state=49)
model_dict['AdaBoost Classifier'] = AdaBoostClassifier(random_state=49,algorithm='SAMME')
model_dict['Gradient Boosting Classifier'] = GradientBoostingClassifier(random_state=49)
```

```
#best accuracy is Adaboost
abc = AdaBoostClassifier(random_state=99)
base = DecisionTreeClassifier(max_depth=3)
abc.fit(X_train,y_train)
pred_abc = abc.predict(X_test)
print('Accuracy of AdaBoost=',accuracy_score(y_test,pred_abc))
```

```
#printing Accuracy of each model
def model_test(X_train, X_test, y_train, y_test,model,model_name):
    model.fit(X_train,y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test,y_pred)
    print('=========================================================={}=========================================================='.format(model_name))
    print('Score is : {}'.format(accuracy))

    print()
```

Classification Reports (Estimated from Accuracy)

Model	Classification Report	F1 Score	Confusion Matrix

Random Forest						75%	[[141 43] [51 141]]
		precision	recall	f1-score	support		
	0	0.73	0.77	0.75	184		
	1	0.77	0.73	0.75	192		
	accuracy			0.75	376		
	macro avg	0.75	0.75	0.75	376		
	weighted avg	0.75	0.75	0.75	376		

Decision Tree						79%	`confusion_matrix(y_test,ypred)` array([[62, 13], [23, 71]])
		precision	recall	f1-score	support		
	0	0.68	0.66	0.67	184		
	1	0.68	0.70	0.69	192		
	accuracy			0.68	376		
	macro avg	0.68	0.68	0.68	376		
	weighted avg	0.68	0.68	0.68	376		

KNN						64%	[[128 56] [68 124]]
		precision	recall	f1-score	support		
	0	0.65	0.70	0.67	184		
	1	0.69	0.65	0.67	192		
	accuracy			0.67	376		
	macro avg	0.67	0.67	0.67	376		
	weighted avg	0.67	0.67	0.67	376		

Gradient Boosting		precision	recall	f1-score	support	78%	[[140 44] [53 139]]
	0	0.73	0.76	0.74	184		
	1	0.76	0.72	0.74	192		
	accuracy			0.74	376		
	macro avg	0.74	0.74	0.74	376		
	weighted avg	0.74	0.74	0.74	376		