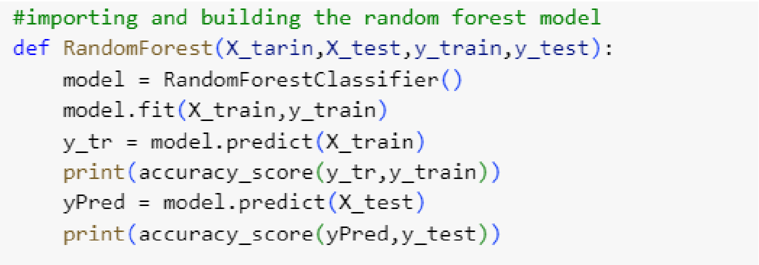
Model Development Phase Template

Date	09 July 2024
Team ID	739801
Project Title	Sepsis Survival Minimal Clinical Records
Maximum Marks	4 Marks

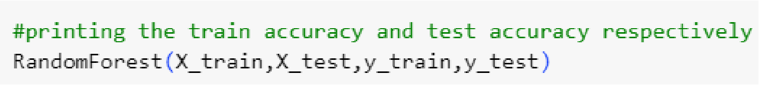
Initial Model Training Code, Model Validation and Evaluation Report

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.

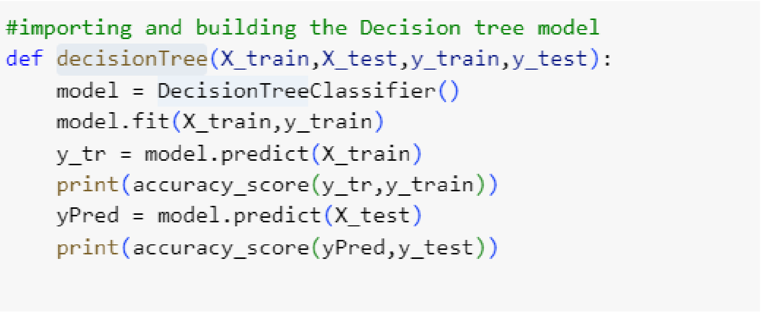
Initial Model Training Code:



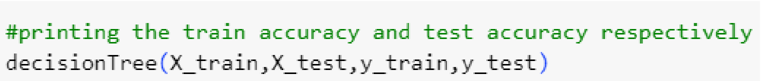
```
#importing and building the random forest model
def RandomForest(X_tarin,X_test,y_train,y_test):
    model = RandomForestClassifier()
    model.fit(X_train,y_train)
    y_tr = model.predict(X_train)
    print(accuracy_score(y_tr,y_train))
    yPred = model.predict(X_test)
    print(accuracy_score(yPred,y_test))
```



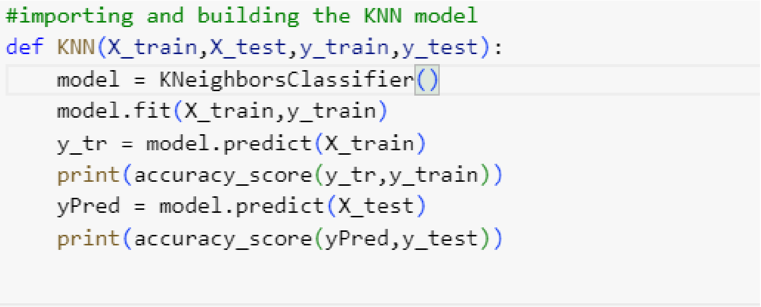
```
#printing the train accuracy and test accuracy respectively
RandomForest(X_train,X_test,y_train,y_test)
```



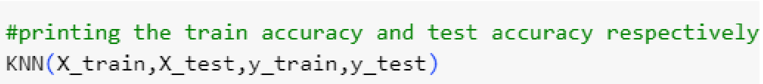
```
#importing and building the Decision tree model
def decisionTree(X_train,X_test,y_train,y_test):
    model = DecisionTreeClassifier()
    model.fit(X_train,y_train)
    y_tr = model.predict(X_train)
    print(accuracy_score(y_tr,y_train))
    yPred = model.predict(X_test)
    print(accuracy_score(yPred,y_test))
```



```
#printing the train accuracy and test accuracy respectively
decisionTree(X_train,X_test,y_train,y_test)
```

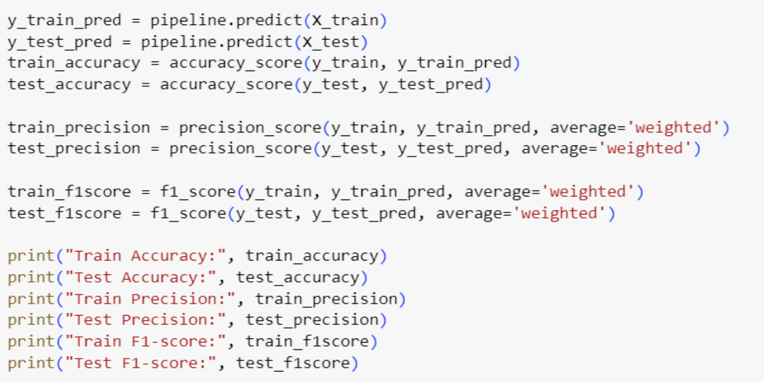


```
#importing and building the KNN model
def KNN(X_train,X_test,y_train,y_test):
    model = KNeighborsClassifier()
    model.fit(X_train,y_train)
    y_tr = model.predict(X_train)
    print(accuracy_score(y_tr,y_train))
    yPred = model.predict(X_test)
    print(accuracy_score(yPred,y_test))
```



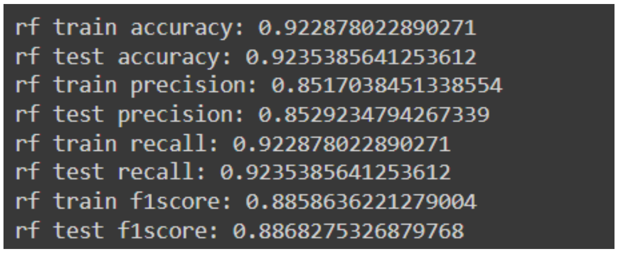
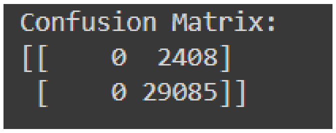
```
#printing the train accuracy and test accuracy respectively
KNN(X_train,X_test,y_train,y_test)
```

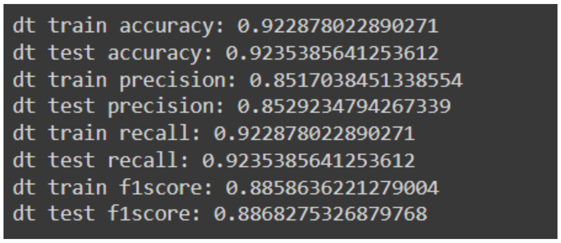
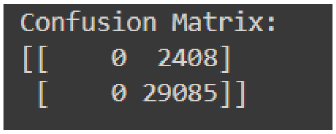
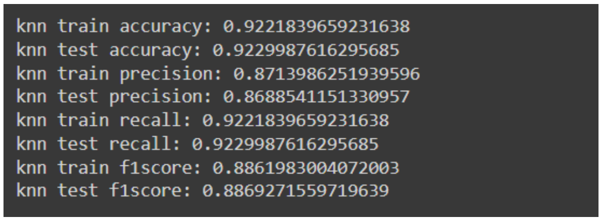
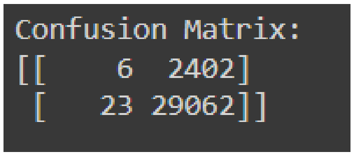
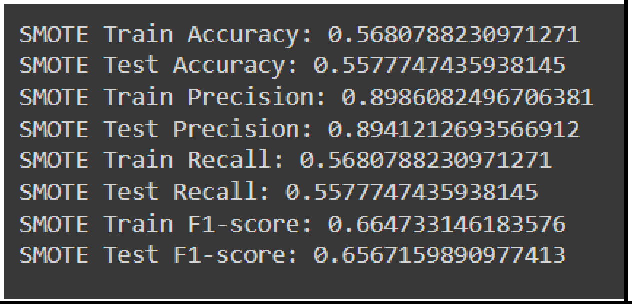
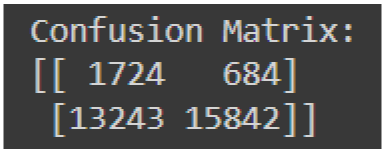
SMOTE



```
y_train_pred = pipeline.predict(X_train)
y_test_pred = pipeline.predict(X_test)
train_accuracy = accuracy_score(y_train, y_train_pred)
test_accuracy = accuracy_score(y_test, y_test_pred)

train_precision = precision_score(y_train, y_train_pred, average='weighted')
test_precision = precision_score(y_test, y_test_pred, average='weighted')

train_f1score = f1_score(y_train, y_train_pred, average='weighted')
test_f1score = f1_score(y_test, y_test_pred, average='weighted')

print("Train Accuracy:", train_accuracy)
print("Test Accuracy:", test_accuracy)
print("Train Precision:", train_precision)
print("Test Precision:", test_precision)
print("Train F1-score:", train_f1score)
print("Test F1-score:", test_f1score)
```

Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix
Random Forest	 rf train accuracy: 0.922878022890271 rf test accuracy: 0.9235385641253612 rf train precision: 0.8517038451338554 rf test precision: 0.8529234794267339 rf train recall: 0.922878022890271 rf test recall: 0.9235385641253612 rf train f1score: 0.8858636221279004 rf test f1score: 0.8868275326879768	88%	 Confusion Matrix: [[0 2408] [0 29085]]

Decision Tree	 dt train accuracy: 0.922878022890271 dt test accuracy: 0.9235385641253612 dt train precision: 0.8517038451338554 dt test precision: 0.8529234794267339 dt train recall: 0.922878022890271 dt test recall: 0.9235385641253612 dt train f1score: 0.8858636221279004 dt test f1score: 0.8868275326879768	88%	 Confusion Matrix: [[0 2408] [0 29085]]
KNN	 knn train accuracy: 0.9221839659231638 knn test accuracy: 0.9229987616295685 knn train precision: 0.8713986251939596 knn test precision: 0.8688541151330957 knn train recall: 0.9221839659231638 knn test recall: 0.9229987616295685 knn train f1score: 0.8861983004072003 knn test f1score: 0.8869271559719639	88%	 Confusion Matrix: [[6 2402] [23 29062]]
SMOTE	 SMOTE Train Accuracy: 0.5680788230971271 SMOTE Test Accuracy: 0.5577747435938145 SMOTE Train Precision: 0.8986082496706381 SMOTE Test Precision: 0.8941212693566912 SMOTE Train Recall: 0.5680788230971271 SMOTE Test Recall: 0.5577747435938145 SMOTE Train F1-score: 0.664733146183576 SMOTE Test F1-score: 0.6567159890977413	65%	 Confusion Matrix: [[1724 684] [13243 15842]]