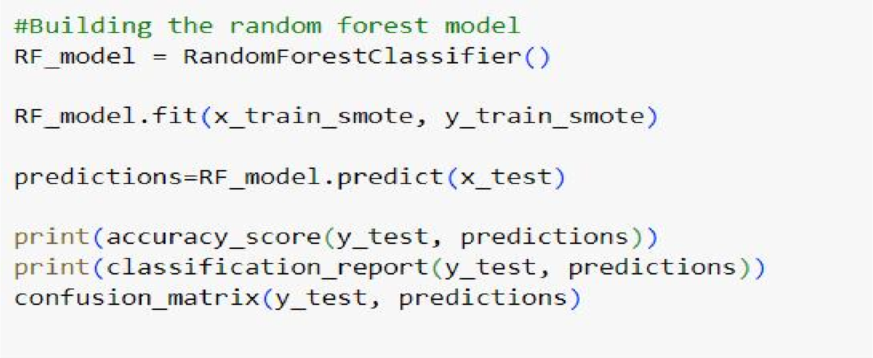
Model Development Phase Template

Date	1 May 2024
Team ID	Team - 737850
Project Title	FetalAI: Using Machine Learning To Predict And Monitor Fetal Health
Maximum Marks	4 Marks

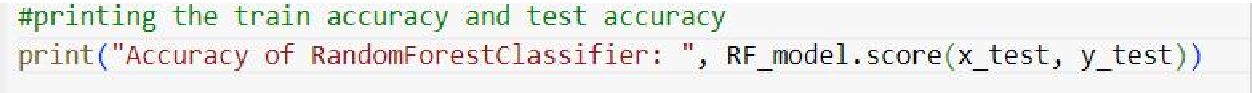
Initial Model Training Code, Model Validation and Evaluation Report

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.

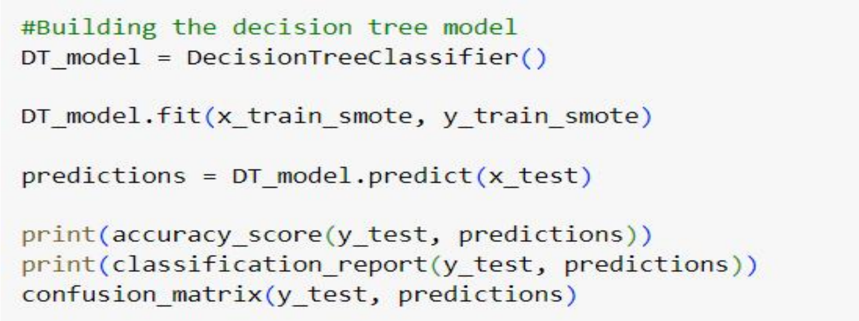
Initial Model Training Code:



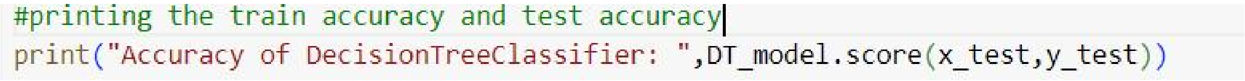
```
#Building the random forest model
RF_model = RandomForestClassifier()

RF_model.fit(x_train_smote, y_train_smote)

predictions=RF_model.predict(x_test)

print(accuracy_score(y_test, predictions))
print(classification_report(y_test, predictions))
confusion_matrix(y_test, predictions)
```



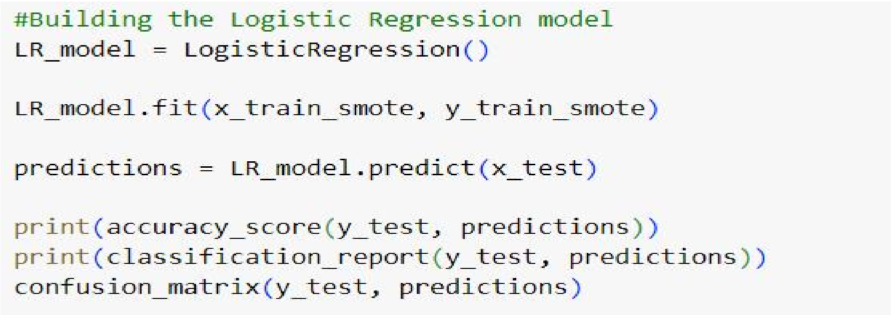
```
#printing the train accuracy and test accuracy
print("Accuracy of RandomForestClassifier: ", RF_model.score(x_test, y_test))
```



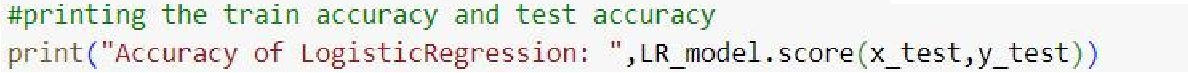
```
#Building the decision tree model
DT_model = DecisionTreeClassifier()

DT_model.fit(x_train_smote, y_train_smote)

predictions = DT_model.predict(x_test)

print(accuracy_score(y_test, predictions))
print(classification_report(y_test, predictions))
confusion_matrix(y_test, predictions)
```



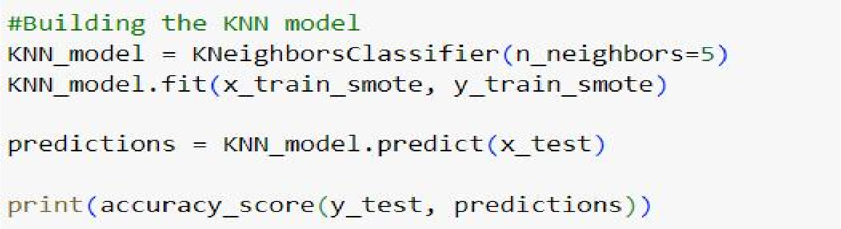
```
#printing the train accuracy and test accuracy
print("Accuracy of DecisionTreeClassifier: ",DT_model.score(x_test,y_test))
```



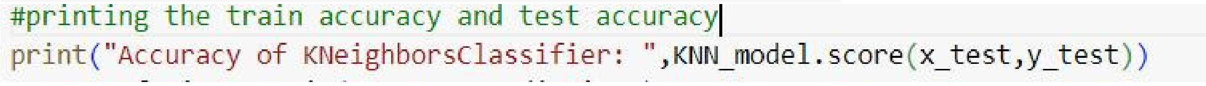
```
#Building the Logistic Regression model
LR_model = LogisticRegression()

LR_model.fit(x_train_smote, y_train_smote)

predictions = LR_model.predict(x_test)

print(accuracy_score(y_test, predictions))
print(classification_report(y_test, predictions))
confusion_matrix(y_test, predictions)
```



```
#printing the train accuracy and test accuracy
print("Accuracy of LogisticRegression: ",LR_model.score(x_test,y_test))
```



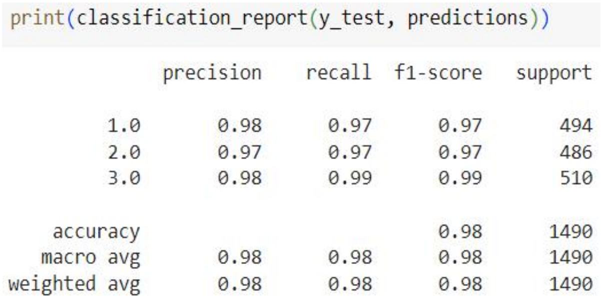
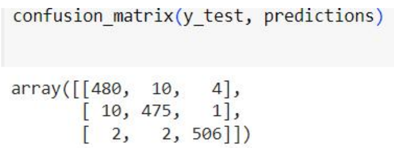
```
#Building the KNN model
KNN_model = KNeighborsClassifier(n_neighbors=5)
KNN_model.fit(x_train_smote, y_train_smote)

predictions = KNN_model.predict(x_test)

print(accuracy_score(y_test, predictions))
```

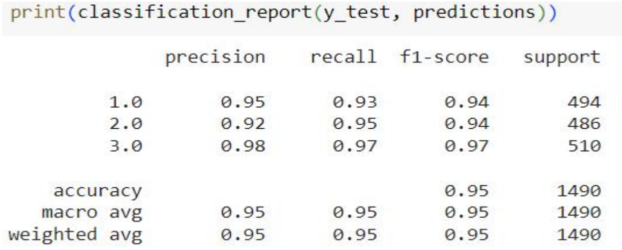
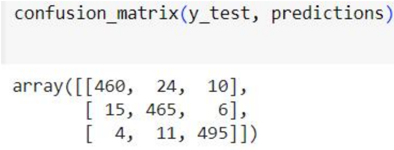
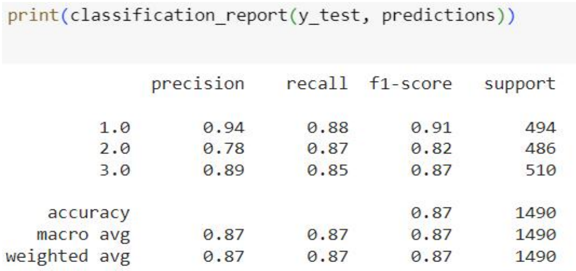
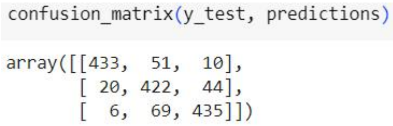
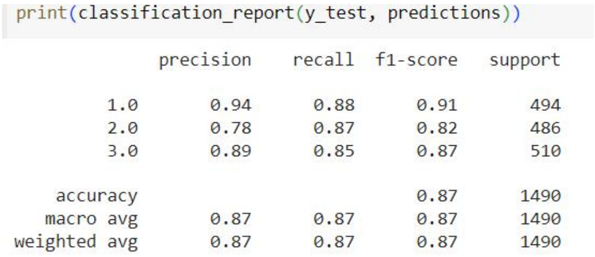
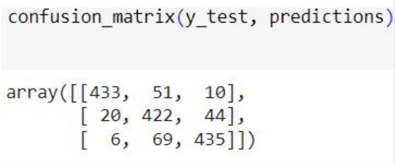


```
#printing the train accuracy and test accuracy
print("Accuracy of KNeighborsClassifier: ",KNN_model.score(x_test,y_test))
```

Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix

Random Forest	 `print(classification_report(y_test, predictions))`
	`precision recall f1-score support`
	`1.0 0.98 0.97 0.97 494`
	`2.0 0.97 0.97 0.97 486`
	`3.0 0.98 0.99 0.99 510`
	`accuracy 0.98 1490`
	`macro avg 0.98 0.98 0.98 1490`
	`weighted avg 0.98 0.98 0.98 1490`
		98%	 `confusion_matrix(y_test, predictions)`
			`array([[480, 10, 4],`
			`[ 10, 475, 1],`
			`[ 2, 2, 506]])`

Decision Tree	 print(classification_report(y_test, predictions)) precision recall f1-score support 1.0 0.95 0.93 0.94 494 2.0 0.92 0.95 0.94 486 3.0 0.98 0.97 0.97 510 accuracy 0.95 1490 macro avg 0.95 0.95 0.95 1490 weighted avg 0.95 0.95 0.95 1490	95%	 confusion_matrix(y_test, predictions) array([[460, 24, 10], [15, 465, 6], [4, 11, 495]])
KNN	 print(classification_report(y_test, predictions)) precision recall f1-score support 1.0 0.94 0.88 0.91 494 2.0 0.78 0.87 0.82 486 3.0 0.89 0.85 0.87 510 accuracy 0.87 1490 macro avg 0.87 0.87 0.87 1490 weighted avg 0.87 0.87 0.87 1490	87%	 confusion_matrix(y_test, predictions) array([[433, 51, 10], [20, 422, 44], [6, 69, 435]])
Logistic Regression	 print(classification_report(y_test, predictions)) precision recall f1-score support 1.0 0.94 0.88 0.91 494 2.0 0.78 0.87 0.82 486 3.0 0.89 0.85 0.87 510 accuracy 0.87 1490 macro avg 0.87 0.87 0.87 1490 weighted avg 0.87 0.87 0.87 1490	87%	 confusion_matrix(y_test, predictions) array([[433, 51, 10], [20, 422, 44], [6, 69, 435]])