Model Development Phase Template

Date	27th July 2024
Team ID	739919
Project Title	FETAL AI: USING MACHINE LEARNING TO PREDICT AND MONITOR FETAL HEALTH
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

In The initial model training for Fetal AI involved using a dataset of fetal health indicators to train a machine learning model, optimizing parameters to maximize accuracy and predictive power. Validation was conducted using a holdout dataset to assess the model's generalizability, while evaluation metrics such as precision, recall, and F1-score were calculated to ensure robust performance. This approach ensures the model effectively identifies critical fetal health patterns, supporting timely and accurate clinical decision-making.

Initial Model Training Code:

```
#Random forest model
from sklearn.ensemble import RandomForestClassifier
RF_model = RandomForestClassifier()
RF_model.fit(X_train_smote,y_train_smote)
predictions = RF_model.predict(X_test)
print(accuracy_score(y_test, predictions))
```

```
0.95141065830721
```

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
import matplotlib.pyplot as plt
```

Empty markdown cell, double-click or press enter to edit.

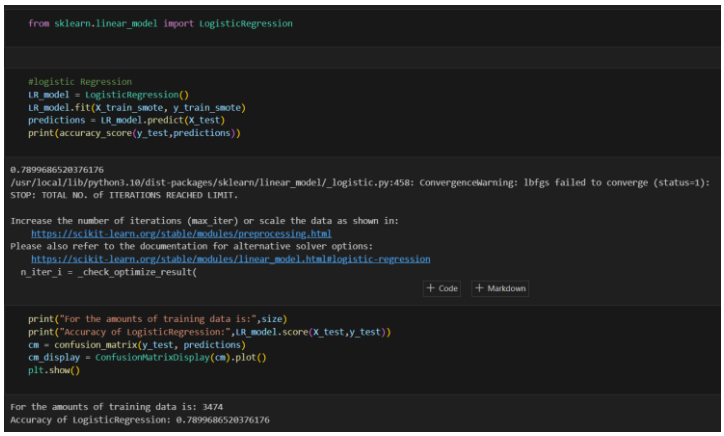
```
size = X_train_smote.shape[0];
print("For the amounts of training data is:",size)
print("accuracy of the RandomForestClassifier:",RF_model.score(X_test,y_test))
cm = confusion_matrix(y_test, predictions)
cm_display = ConfusionMatrixDisplay(cm).plot()
plt.show()
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix

Random Forest Regressor	`#Random forest model`
	`from sklearn.ensemble import RandomForestClassifier`
	`RF_model = RandomForestClassifier()`
	`RF_model.fit(X_train_smote,y_train_smote)`
	`predictions = RF_model.predict(X_test)`
	`print(accuracy_score(y_test, predictions))`
	`0.95141065830721`
	`from sklearn.metrics import confusion_matrix`
	`from sklearn.metrics import ConfusionMatrixDisplay`
	`import matplotlib.pyplot as plt`
	Empty markdown cell, double-click or press enter to edit.
	`size = X_train_smote.shape[0];`
	`print("For the amounts of training data is:",size)`
	`print("accuracy of the RandomForestClassifier:",RF_model.score(X_test,y_test))`
	`cm = confusion_matrix(y_test, predictions)`
	`cm_display = ConfusionMatrixDisplay(cm).plot()`
	`plt.show()`
		95%	-

Decision Tree Regressor	`from sklearn.tree import DecisionTreeClassifier`
	`#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))`
	`0.9184952978056427`
	`print("For the amounts of training data is:",size)`
	`print("Accuracy of DecisionTreeClassifier:",DT_model.score(X_test,y_test))`
	`cm = confusion_matrix(y_test, predictions)`
	`cm_display = ConfusionMatrixDisplay(cm).plot()`
	`plt.show()`
	`For the amounts of training data is: 3474`
	`Accuracy of DecisionTreeClassifier: 0.9184952978056427`
		91%	-

Logistic regression		78%	-