

Model Development Phase Template

Date	10 JULY 2024
Team ID	FACULTY
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

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:

Random Forest Model RF

```

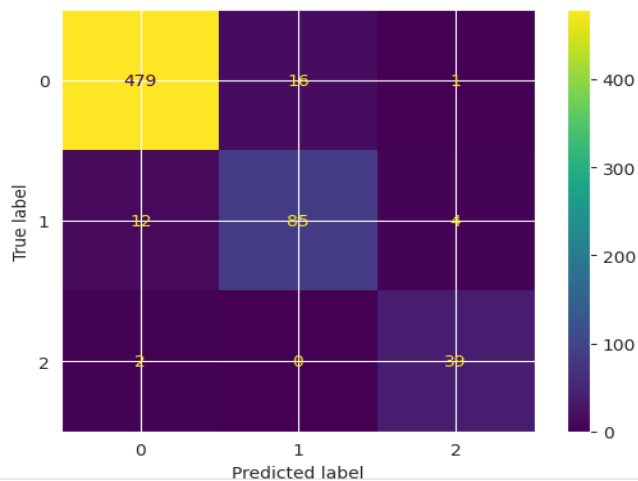
[30] RF_model=RandomForestClassifier()
      RF_model.fit(x_train_smote,y_train_smote)
      predictions=RF_model.predict(x_test)
      print(accuracy_score(y_test,predictions))
  
```

0.945141065830721

```

[31] print("For the amounts of training data is:", x_train_smote.shape[0]) # Assu
      print("Accuracy of RandomForestClassifier:", RF_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: 3477
Accuracy of RandomForestClassifier: 0.945141065830721



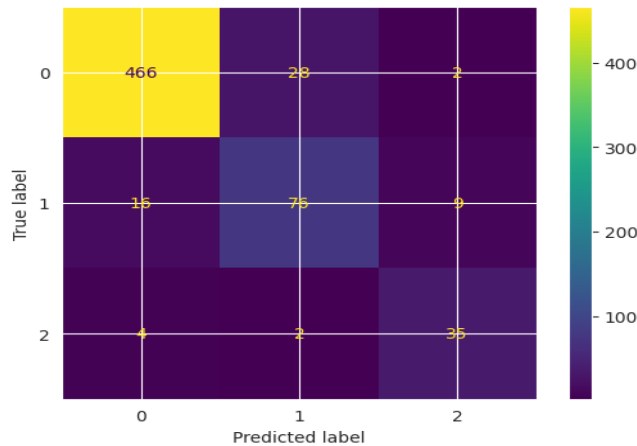
Decision Tree DT

```
[32] DT_model=DecisionTreeClassifier()
DT_model.fit(x_train_smote,y_train_smote)
predictions=DT_model.predict(x_test)
print(accuracy_score(y_test,predictions))
```

0.9043887147335423

```
[33] print("For the amounts of training data is:",x_train_smote.shape[0])
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: 3477
Accuracy of DecisionTreeClassifier: 0.9043887147335423



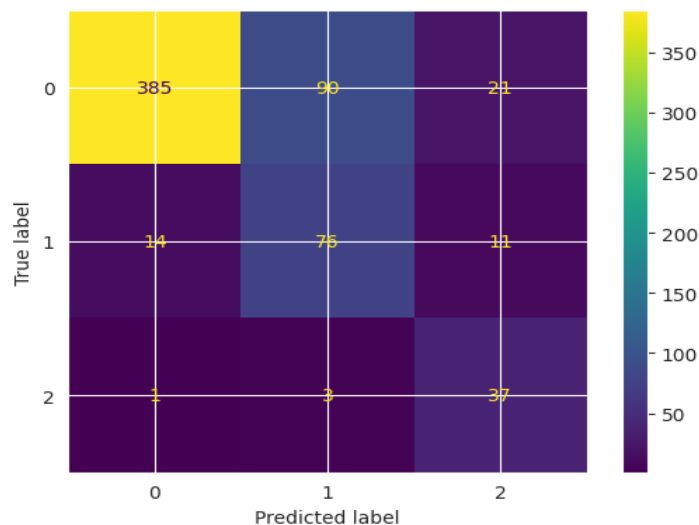
[Logistic Regression] LR

```
[34] LR_model=LogisticRegression()
LR_model.fit(x_train_smote,y_train_smote)
predictions=LR_model.predict(x_test)
print(accuracy_score(y_test,predictions))
```

0.780564263322884

```
[35] print("For the amounts of training data is:",x_train_smote.shape[0])
print("Accuracy of LogisticRegression:",LR_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: 3477
Accuracy of LogisticRegression: 0.780564263322884



KNN K-Nearest Neighbors

```

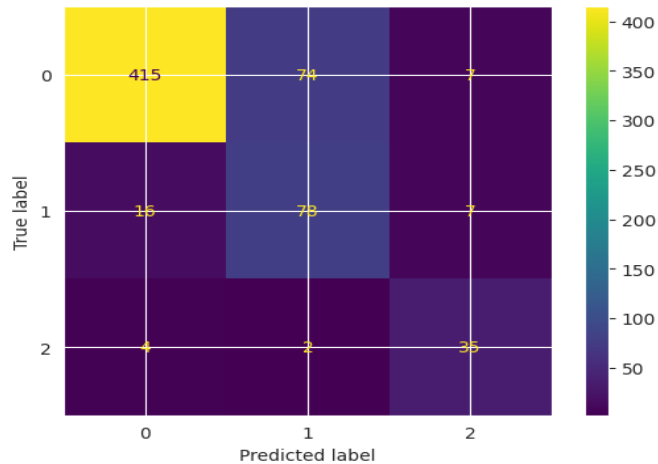
[36] KNN_model=KNeighborsClassifier()
      KNN_model.fit(x_train_smote,y_train_smote)
      predictions=KNN_model.predict(x_test)
      print(accuracy_score(y_test,predictions))
  
```

0.8275862068965517

```

[37] print("For the amounts of training data is:",x_train_smote.shape[0])
      print("Accuracy of KNeighborsClassifier:",KNN_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: 3477
Accuracy of KNeighborsClassifier: 0.8275862068965517



Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix																																	
Random Forest	<div><div><div>0s</div><div><pre>print(classification_report(y_test,predictions))</pre></div><div></div></div><table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>1.0</td><td>0.96</td><td>0.84</td><td>0.90</td><td>496</td></tr><tr><td>2.0</td><td>0.53</td><td>0.81</td><td>0.64</td><td>101</td></tr><tr><td>3.0</td><td>0.71</td><td>0.83</td><td>0.76</td><td>41</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.84</td><td>638</td></tr><tr><td>macro avg</td><td>0.73</td><td>0.83</td><td>0.77</td><td>638</td></tr><tr><td>weighted avg</td><td>0.88</td><td>0.84</td><td>0.85</td><td>638</td></tr></tbody></table></div> <div>93%</div> <div><div><div>0s</div><div><pre>[36] confusion_matrix(y_test,predictions)</pre></div><div></div></div><div><pre>array([[477, 17, 2], [15, 82, 4], [2, 0, 39]])</pre></div></div>		precision	recall	f1-score	support	1.0	0.96	0.84	0.90	496	2.0	0.53	0.81	0.64	101	3.0	0.71	0.83	0.76	41	accuracy			0.84	638	macro avg	0.73	0.83	0.77	638	weighted avg	0.88	0.84	0.85	638
	precision	recall	f1-score	support																																
1.0	0.96	0.84	0.90	496																																
2.0	0.53	0.81	0.64	101																																
3.0	0.71	0.83	0.76	41																																
accuracy			0.84	638																																
macro avg	0.73	0.83	0.77	638																																
weighted avg	0.88	0.84	0.85	638																																
Decision Tree	<div><div><div>0s</div><div><pre>[39] print(classification_report(y_test,predictions))</pre></div><div></div></div><table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>1.0</td><td>0.97</td><td>0.93</td><td>0.95</td><td>496</td></tr><tr><td>2.0</td><td>0.73</td><td>0.80</td><td>0.76</td><td>101</td></tr><tr><td>3.0</td><td>0.73</td><td>0.88</td><td>0.80</td><td>41</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.91</td><td>638</td></tr><tr><td>macro avg</td><td>0.81</td><td>0.87</td><td>0.84</td><td>638</td></tr><tr><td>weighted avg</td><td>0.91</td><td>0.91</td><td>0.91</td><td>638</td></tr></tbody></table></div> <div>90%</div> <div><div><div>0s</div><div><pre>[40] confusion_matrix(y_test,predictions)</pre></div><div></div></div><div><pre>array([[462, 29, 5], [12, 81, 8], [4, 1, 36]])</pre></div></div>		precision	recall	f1-score	support	1.0	0.97	0.93	0.95	496	2.0	0.73	0.80	0.76	101	3.0	0.73	0.88	0.80	41	accuracy			0.91	638	macro avg	0.81	0.87	0.84	638	weighted avg	0.91	0.91	0.91	638
	precision	recall	f1-score	support																																
1.0	0.97	0.93	0.95	496																																
2.0	0.73	0.80	0.76	101																																
3.0	0.73	0.88	0.80	41																																
accuracy			0.91	638																																
macro avg	0.81	0.87	0.84	638																																
weighted avg	0.91	0.91	0.91	638																																
Logistic Regression	<div><div><div>0s</div><div><pre>[43] print(classification_report(y_test,predictions))</pre></div><div></div></div><table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>1.0</td><td>0.97</td><td>0.77</td><td>0.86</td><td>496</td></tr><tr><td>2.0</td><td>0.44</td><td>0.73</td><td>0.55</td><td>101</td></tr><tr><td>3.0</td><td>0.49</td><td>0.88</td><td>0.63</td><td>41</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.77</td><td>638</td></tr><tr><td>macro avg</td><td>0.63</td><td>0.79</td><td>0.68</td><td>638</td></tr><tr><td>weighted avg</td><td>0.85</td><td>0.77</td><td>0.79</td><td>638</td></tr></tbody></table></div> <div>77%</div> <div><div><div></div><div><pre>[44] confusion_matrix(y_test,predictions)</pre></div><div></div></div><div><pre>array([[382, 92, 22], [12, 74, 15], [1, 4, 36]])</pre></div></div>		precision	recall	f1-score	support	1.0	0.97	0.77	0.86	496	2.0	0.44	0.73	0.55	101	3.0	0.49	0.88	0.63	41	accuracy			0.77	638	macro avg	0.63	0.79	0.68	638	weighted avg	0.85	0.77	0.79	638
	precision	recall	f1-score	support																																
1.0	0.97	0.77	0.86	496																																
2.0	0.44	0.73	0.55	101																																
3.0	0.49	0.88	0.63	41																																
accuracy			0.77	638																																
macro avg	0.63	0.79	0.68	638																																
weighted avg	0.85	0.77	0.79	638																																
KNN KNeighbors Classifier	<div><div><div>0s</div><div><pre>[47] print(classification_report(y_test,predictions))</pre></div><div></div></div><table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>1.0</td><td>0.96</td><td>0.84</td><td>0.90</td><td>496</td></tr><tr><td>2.0</td><td>0.53</td><td>0.81</td><td>0.64</td><td>101</td></tr><tr><td>3.0</td><td>0.71</td><td>0.83</td><td>0.76</td><td>41</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.84</td><td>638</td></tr><tr><td>macro avg</td><td>0.73</td><td>0.83</td><td>0.77</td><td>638</td></tr><tr><td>weighted avg</td><td>0.88</td><td>0.84</td><td>0.85</td><td>638</td></tr></tbody></table></div> <div>83%</div> <div><div><div>0s</div><div><pre>[48] confusion_matrix(y_test,predictions)</pre></div><div></div></div><div><pre>array([[418, 69, 9], [14, 82, 5], [4, 3, 34]])</pre></div></div>		precision	recall	f1-score	support	1.0	0.96	0.84	0.90	496	2.0	0.53	0.81	0.64	101	3.0	0.71	0.83	0.76	41	accuracy			0.84	638	macro avg	0.73	0.83	0.77	638	weighted avg	0.88	0.84	0.85	638
	precision	recall	f1-score	support																																
1.0	0.96	0.84	0.90	496																																
2.0	0.53	0.81	0.64	101																																
3.0	0.71	0.83	0.76	41																																
accuracy			0.84	638																																
macro avg	0.73	0.83	0.77	638																																
weighted avg	0.88	0.84	0.85	638																																