

## Model Development Phase Template

Date	21 July 2024
Team ID	Team-740025
Project Title	Unlocking Silent Signals :Decoding Body Language with Mediapipe
Maximum Marks	10 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 a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

### Initial Model Training Code (5 marks):

Paste the screenshot of the model training code

### Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
-------	---------	---

Model 1	<p>GradientBoostingClassifier model typically include accuracy, precision, recall, F1 score to evaluate its predictive performance and generalization capability.</p>	<pre>from sklearn.ensemble import GradientBoostingClassifier #train gbc = GradientBoostingClassifier(learning_rate=0.02,                                 max_depth=4,                                 random_state=100, n_estimators=1000) gbc.fit(X_train,y_train) #predict y_predicted_gb = gbc.predict(X_test) print("Training Accuracy :", gbc.score(X_train, y_train)) print("Testing Accuracy :", gbc.score(X_test, y_test)) #eval #cm = confusion_matrix(y_test, y_predicted_gb) #plt.rcParams['figure.figsize'] = (3, 3) #sns.heatmap(cm, annot = True, cmap = 'YlGnBu', fmt = '.8g') #plt.show() cr = classification_report(y_test, y_predicted_gb) print(cr) print("-----") false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test,y_predicted_gb) roc_auc = auc(false_positive_rate, true_positive_rate) print("ROC Curves",roc_auc) precision, recall, thresholds = precision_recall_curve(y_test, y_predicted_gb) f1 = f1_score(y_test, y_predicted_gb) Precision_Recall_gbs = auc(recall, precision) print("Precision-Recall Curves =",Precision_Recall_gbs) ✓ 41s</pre> <p>Training Accuracy : 0.9984520123839009 Testing Accuracy : 0.740072202166065</p> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.65</td><td>0.57</td><td>0.61</td><td>98</td></tr><tr><td>1</td><td>0.78</td><td>0.83</td><td>0.81</td><td>179</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.74</td><td>277</td></tr><tr><td>macro avg</td><td>0.72</td><td>0.70</td><td>0.71</td><td>277</td></tr></table>		precision	recall	f1-score	support	0	0.65	0.57	0.61	98	1	0.78	0.83	0.81	179	accuracy			0.74	277	macro avg	0.72	0.70	0.71	277					
	precision	recall	f1-score	support																												
0	0.65	0.57	0.61	98																												
1	0.78	0.83	0.81	179																												
accuracy			0.74	277																												
macro avg	0.72	0.70	0.71	277																												
Model 2	<p>AdaBoost classifier model commonly include accuracy, precision, recall, F1 score which help assess the model's prediction accuracy and generalizability</p>	<pre>from sklearn.ensemble import AdaBoostClassifier #train ada = AdaBoostClassifier() ada.fit(X_train,y_train) #predict y_predicted_ab = ada.predict(X_test) print("Training Accuracy :", ada.score(X_train, y_train)) print("Testing Accuracy :", ada.score(X_test, y_test)) cr = classification_report(y_test, y_predicted_ab) print(cr) false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test,y_predicted_ab) roc_auc = auc(false_positive_rate, true_positive_rate) print("roc_auc",roc_auc) print("-----") false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test,y_predicted_ab) roc_auc = auc(false_positive_rate, true_positive_rate) print("ROC Curves",roc_auc) precision, recall, thresholds = precision_recall_curve(y_test, y_predicted_ab) f1 = f1_score(y_test, y_predicted_ab) Precision_Recall_abs = auc(recall, precision) print("Precision-Recall Curves =",Precision_Recall_abs) ✓ 02s</pre> <p>Training Accuracy : 0.8328173374613003 Testing Accuracy : 0.776173285198556</p> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.72</td><td>0.60</td><td>0.66</td><td>98</td></tr><tr><td>1</td><td>0.80</td><td>0.87</td><td>0.83</td><td>179</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.78</td><td>277</td></tr><tr><td>macro avg</td><td>0.76</td><td>0.74</td><td>0.74</td><td>277</td></tr><tr><td>weighted avg</td><td>0.77</td><td>0.78</td><td>0.77</td><td>277</td></tr></table>		precision	recall	f1-score	support	0	0.72	0.60	0.66	98	1	0.80	0.87	0.83	179	accuracy			0.78	277	macro avg	0.76	0.74	0.74	277	weighted avg	0.77	0.78	0.77	277
	precision	recall	f1-score	support																												
0	0.72	0.60	0.66	98																												
1	0.80	0.87	0.83	179																												
accuracy			0.78	277																												
macro avg	0.76	0.74	0.74	277																												
weighted avg	0.77	0.78	0.77	277																												

Model 3	<p>Random forest classifier model often encompass accuracy, precision, recall, F1 score to measure its prediction quality and robustness.</p>	<pre>from sklearn.ensemble import RandomForestClassifier rf = RandomForestClassifier() rf.fit(X_train.get_numeric_data(), y_train) y_pred_rf = rf.predict(X_test.get_numeric_data()) print("Training Accuracy :", rf.score(X_train.get_numeric_data(), y_train)) print("Testing Accuracy :", rf.score(X_test.get_numeric_data(), y_test)) cm = confusion_matrix(y_test, y_pred_rf) plt.rcParams['figure.figsize'] = (3, 3) sns.heatmap(cm, annot = True, cmap = 'YlGnBu', fmt = '.8g') plt.show() cr = classification_report(y_test, y_pred_rf) print(cr) print("-----") false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, y_pred_rf) roc_auc = auc(false_positive_rate, true_positive_rate) print("ROC Curves = ", roc_auc) precision, recall, thresholds = precision_recall_curve(y_test, y_pred_rf) f1 = f1_score(y_test, y_pred_rf) Precision_Recall_rfs = auc(recall, precision) print("Precision-Recall Curves = ", Precision_Recall_rfs)</pre> <p>✓ 0.4s</p> <p>Training Accuracy : 1.0 Testing Accuracy : 0.7978339350180506</p> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.76</td><td>0.62</td><td>0.69</td><td>98</td></tr><tr><td>1</td><td>0.81</td><td>0.89</td><td>0.85</td><td>179</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.80</td><td>277</td></tr><tr><td>macro avg</td><td>0.79</td><td>0.76</td><td>0.77</td><td>277</td></tr><tr><td>weighted avg</td><td>0.79</td><td>0.80</td><td>0.79</td><td>277</td></tr></tbody></table> <p>-----</p>		precision	recall	f1-score	support	0	0.76	0.62	0.69	98	1	0.81	0.89	0.85	179	accuracy			0.80	277	macro avg	0.79	0.76	0.77	277	weighted avg	0.79	0.80	0.79	277
	precision	recall	f1-score	support																												
0	0.76	0.62	0.69	98																												
1	0.81	0.89	0.85	179																												
accuracy			0.80	277																												
macro avg	0.79	0.76	0.77	277																												
weighted avg	0.79	0.80	0.79	277																												
Model 4	<p>XGB Classifier model typically include accuracy, precision, recall, F1 score to evaluate its prediction performance and generalization ability</p>	<pre>from xgboost import XGBClassifier #train xgb = XGBClassifier() xgb.fit(X_train, y_train) #predict y_predicted_xgb = xgb.predict(X_test) print("Training Accuracy :", xgb.score(X_train, y_train)) print("Testing Accuracy :", xgb.score(X_test, y_test)) cr = classification_report(y_test, y_predicted_xgb) print(cr) print("-----") false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, y_predicted_xgb) roc_auc = auc(false_positive_rate, true_positive_rate) print("ROC Curves = ", roc_auc) precision, recall, thresholds = precision_recall_curve(y_test, y_predicted_xgb) f1 = f1_score(y_test, y_predicted_xgb) Precision_Recall_xgb = auc(recall, precision) print("Precision-Recall Curves = ", Precision_Recall_xgb)</pre> <p>✓ 1.1s</p> <p>Training Accuracy : 1.0 Testing Accuracy : 0.7653429602888087</p> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.70</td><td>0.58</td><td>0.64</td><td>98</td></tr><tr><td>1</td><td>0.79</td><td>0.87</td><td>0.83</td><td>179</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.77</td><td>277</td></tr><tr><td>macro avg</td><td>0.75</td><td>0.72</td><td>0.73</td><td>277</td></tr><tr><td>weighted avg</td><td>0.76</td><td>0.77</td><td>0.76</td><td>277</td></tr></tbody></table> <p>-----</p> <p>ROC Curves = 0.7237772203853609 Precision-Recall Curves = 0.8716903567590439</p>		precision	recall	f1-score	support	0	0.70	0.58	0.64	98	1	0.79	0.87	0.83	179	accuracy			0.77	277	macro avg	0.75	0.72	0.73	277	weighted avg	0.76	0.77	0.76	277
	precision	recall	f1-score	support																												
0	0.70	0.58	0.64	98																												
1	0.79	0.87	0.83	179																												
accuracy			0.77	277																												
macro avg	0.75	0.72	0.73	277																												
weighted avg	0.76	0.77	0.76	277																												