

Project Development Phase Performance Test

Date	18 November 2022
Team ID	PNT2022TMID37893
Project Name	Project – Real – Time Communication Powered By AI For Specially Abled
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot																																																																																																		
1.	Metrics obtained	<p>Regression Model:</p> <p>Age of ASL – 0.130 Non verbal IQ – 0.519 Exposure –0.354</p> <p>Units consumed during the execution</p> <p>Maximum time taken : 45.3 Minimum time taken : 6.7</p> <p>Classification report</p>	<div><p>ASL syntax</p><table><tr><td>1. Age of ASL exposure</td><td>.130</td><td>.130</td><td>11.704**</td></tr><tr><td>2. Nonverbal IQ</td><td>.391</td><td>.261</td><td>9.663**</td></tr><tr><td>3. Print exposure</td><td>.519</td><td>.128</td><td>7.138**</td></tr></table><p>ASL narrative comprehension</p><table><tr><td>1. Age of ASL exposure</td><td>.032</td><td>.032</td><td>5.823*</td></tr><tr><td>2. Nonverbal IQ</td><td>.354</td><td>.322</td><td>12.367**</td></tr><tr><td>2. Print exposure</td><td>.566</td><td>.212</td><td>13.132**</td></tr></table><p>MCE narrative comprehension</p><table><tr><td>1. Nonverbal IQ</td><td>.302</td><td>.302</td><td>9.922**</td></tr><tr><td>2. Print exposure</td><td>.422</td><td>.120</td><td>5.806*</td></tr></table></div> <div><p>*p < .05. **p < .01.</p></div> <div><pre>In [18]: 1 def some_function(a): 2 a_1 = [v for v in range(a)] 3 a_12 = list(range(a)) 4 a13 = np.arange(0,a) 5 print("done") 6 7 %lprun -f some_function some_function(100)</pre><p>executed in 8ms, finished 15:01:50 2018-06-08</p><p>done</p></div> <div><p>Timer unit: 1e-06 s</p><p>Total time: 0.00015 s</p><p>File: <ipython-input-18-0669e0aafaa9></p><p>Function: some_function at line 1</p><table><tr><th>Line #</th><th>Hits</th><th>Time</th><th>Per Hit</th><th>% Time</th><th>Line Contents</th></tr><tr><td>1</td><td></td><td></td><td></td><td></td><td>def some_function(a):</td></tr><tr><td>2</td><td>1</td><td>27.0</td><td>27.0</td><td>18.0</td><td>a_1 = [v for v in range(a)]</td></tr><tr><td>3</td><td>1</td><td>10.0</td><td>10.0</td><td>6.7</td><td>a_12 = list(range(a))</td></tr><tr><td>4</td><td>1</td><td>45.0</td><td>45.0</td><td>30.0</td><td>a13 = np.arange(0,a)</td></tr><tr><td>5</td><td>1</td><td>68.0</td><td>68.0</td><td>45.3</td><td>print("done")</td></tr></table></div> <div><pre>from sklearn.metrics import classification_report print(classification_report(y_pred , lr.predict(X_test)))</pre><table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>False</td><td>1.00</td><td>1.00</td><td>1.00</td><td>1</td></tr><tr><td>True</td><td>1.00</td><td>1.00</td><td>1.00</td><td>59</td></tr><tr><td>accuracy</td><td></td><td></td><td>1.00</td><td>60</td></tr><tr><td>macro avg</td><td>1.00</td><td>1.00</td><td>1.00</td><td>60</td></tr><tr><td>weighted avg</td><td>1.00</td><td>1.00</td><td>1.00</td><td>60</td></tr></table></div>	1. Age of ASL exposure	.130	.130	11.704**	2. Nonverbal IQ	.391	.261	9.663**	3. Print exposure	.519	.128	7.138**	1. Age of ASL exposure	.032	.032	5.823*	2. Nonverbal IQ	.354	.322	12.367**	2. Print exposure	.566	.212	13.132**	1. Nonverbal IQ	.302	.302	9.922**	2. Print exposure	.422	.120	5.806*	Line #	Hits	Time	Per Hit	% Time	Line Contents	1					def some_function(a):	2	1	27.0	27.0	18.0	a_1 = [v for v in range(a)]	3	1	10.0	10.0	6.7	a_12 = list(range(a))	4	1	45.0	45.0	30.0	a13 = np.arange(0,a)	5	1	68.0	68.0	45.3	print("done")		precision	recall	f1-score	support	False	1.00	1.00	1.00	1	True	1.00	1.00	1.00	59	accuracy			1.00	60	macro avg	1.00	1.00	1.00	60	weighted avg	1.00	1.00	1.00	60
1. Age of ASL exposure	.130	.130	11.704**																																																																																																		
2. Nonverbal IQ	.391	.261	9.663**																																																																																																		
3. Print exposure	.519	.128	7.138**																																																																																																		
1. Age of ASL exposure	.032	.032	5.823*																																																																																																		
2. Nonverbal IQ	.354	.322	12.367**																																																																																																		
2. Print exposure	.566	.212	13.132**																																																																																																		
1. Nonverbal IQ	.302	.302	9.922**																																																																																																		
2. Print exposure	.422	.120	5.806*																																																																																																		
Line #	Hits	Time	Per Hit	% Time	Line Contents																																																																																																
1					def some_function(a):																																																																																																
2	1	27.0	27.0	18.0	a_1 = [v for v in range(a)]																																																																																																
3	1	10.0	10.0	6.7	a_12 = list(range(a))																																																																																																
4	1	45.0	45.0	30.0	a13 = np.arange(0,a)																																																																																																
5	1	68.0	68.0	45.3	print("done")																																																																																																
	precision	recall	f1-score	support																																																																																																	
False	1.00	1.00	1.00	1																																																																																																	
True	1.00	1.00	1.00	59																																																																																																	
accuracy			1.00	60																																																																																																	
macro avg	1.00	1.00	1.00	60																																																																																																	
weighted avg	1.00	1.00	1.00	60																																																																																																	

2.	Model Tuning	Hyper parameter Tuning - Validation Method -	<pre>from numpy.core.numeric import cross from sklearn import datasets from sklearn.linear_model._logistic import LogisticRegression from sklearn.model_selection import StratifiedKFold,cross_val_score X,y = datasets.load_iris(return_X_y=True) lore = LogisticRegression(random_state=0, max_iter=1000) sk_folds = StratifiedKFold(n_splits= 5) scores = cross_val_score(lore,X,y,cv= sk_folds) print("Cross Validation Scores:",scores) print("Average CV Scores:",scores.mean()) print("Number of CV Scores used in Average:",len(scores))</pre> <p>Cross Validation Scores: [0.96666667 1. 0.93333333 0.96666667 1.] Average CV Scores: 0.9733333333333334 Number of CV Scores used in Average: 5</p>
----	--------------	--	--