

**Project Development Phase**  
**Model Performance Test**

Date	10 November 2022
Team ID	PNT2022TMID35730
Project Name	Early Detection of Chronic Kidney Disease using Machine Learning Techniques
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	<b>Classification Model:</b> Confusion Matrix - , Accuracy Score- & Classification Report -	Training Accuracy of Final Ensemble Model is 100.0 Test Accuracy of Final Ensemble Model is 100.0  Confusion Matrix :- [[44 0] [ 0 76]]  Classification Report :- <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>1.00</td><td>1.00</td><td>1.00</td><td>44</td></tr><tr><td>1</td><td>1.00</td><td>1.00</td><td>1.00</td><td>76</td></tr><tr><td>accuracy</td><td></td><td></td><td>1.00</td><td>120</td></tr><tr><td>macro avg</td><td>1.00</td><td>1.00</td><td>1.00</td><td>120</td></tr><tr><td>weighted avg</td><td>1.00</td><td>1.00</td><td>1.00</td><td>120</td></tr></tbody></table>		precision	recall	f1-score	support	0	1.00	1.00	1.00	44	1	1.00	1.00	1.00	76	accuracy			1.00	120	macro avg	1.00	1.00	1.00	120	weighted avg	1.00	1.00	1.00	120
	precision	recall	f1-score	support																													
0	1.00	1.00	1.00	44																													
1	1.00	1.00	1.00	76																													
accuracy			1.00	120																													
macro avg	1.00	1.00	1.00	120																													
weighted avg	1.00	1.00	1.00	120																													

2.	Tune the Model	Grid Search CV	<pre data-bbox="539 338 1469 651">In [18]: # hyper parameter tuning of decision tree  from sklearn.model_selection import GridSearchCV grid_param = {     'criterion' : ['gini', 'entropy'],     'max_depth' : [3, 5, 7, 10],     'splitter' : ['best', 'random'],     'min_samples_leaf' : [1, 2, 3, 5, 7],     'min_samples_split' : [1, 2, 3, 5, 7],     'max_features' : ['auto', 'sqrt', 'log2'] }  grid_search_dtc = GridSearchCV(dtc, grid_param, cv = 5, n_jobs = -1, verbose = 1) grid_search_dtc.fit(X_train, y_train)</pre> <p data-bbox="636 669 1294 692">Fitting 5 folds for each of 1200 candidates, totalling 6000 fits</p> <pre data-bbox="539 710 1313 887">Out[18]: GridSearchCV(cv=5, estimator=DecisionTreeClassifier(), n_jobs=-1,                     param_grid={'criterion': ['gini', 'entropy'],                                 'max_depth': [3, 5, 7, 10],                                 'max_features': ['auto', 'sqrt', 'log2'],                                 'min_samples_leaf': [1, 2, 3, 5, 7],                                 'min_samples_split': [1, 2, 3, 5, 7],                                 'splitter': ['best', 'random']}},                     verbose=1)</pre>
----	----------------	----------------	--