

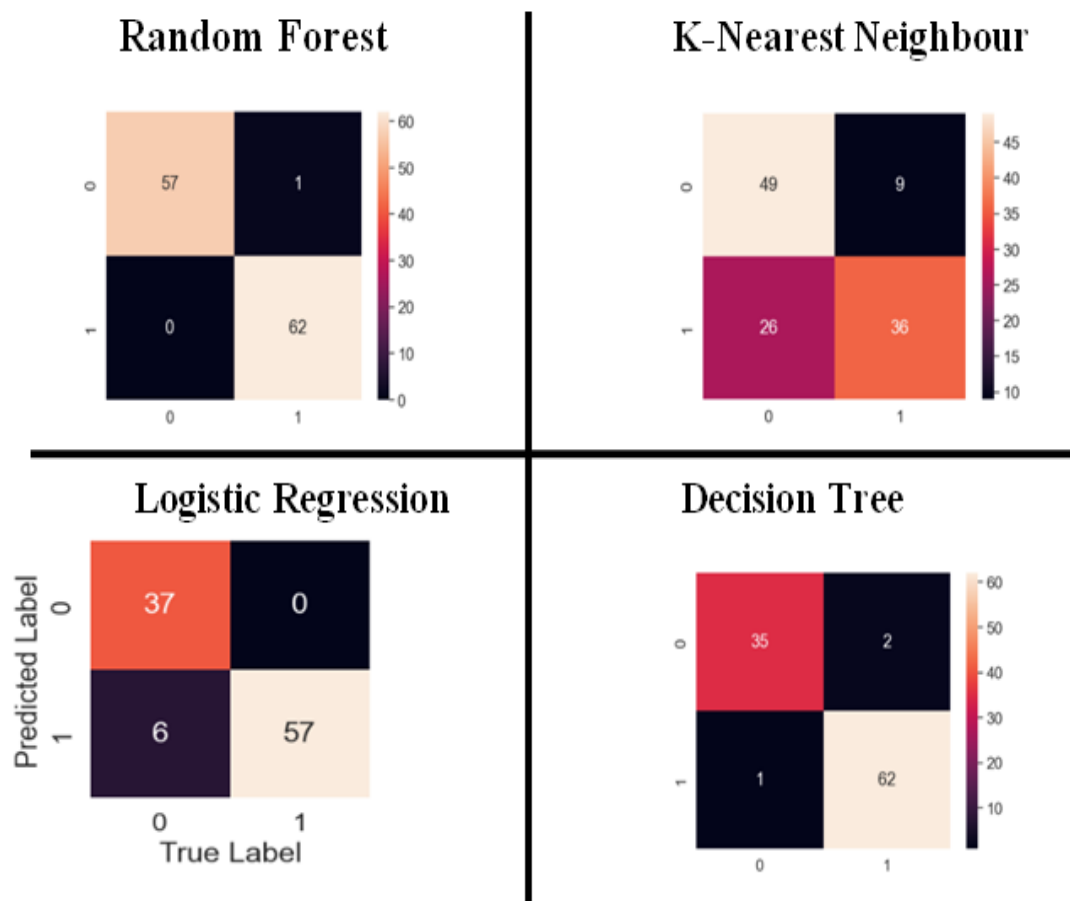
Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID07996
Project Name	Early Detection of Chronic Kidney disease using Machine Learning
Maximum Marks	10 Marks

Model Performance Testing:

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

Confusion matrix:



S.No.	Parameter	Values	Screenshot																																																																																														
1.	Metrics	<p>Regression Model: MAE - , MSE - , RMSE - , R2 score -</p> <p>Classification Model: Confusion Matrix - , Accuracy Score- & Classification Report -</p>	<div><pre>1 from sklearn import metrics 2 print("MAE",metrics.mean_absolute_error(y_test,ypred)) 3 print("MSE",metrics.mean_squared_error(y_test,ypred)) 4 print("RMSE",np.sqrt(metrics.mean_squared_error(y_test,ypred))) 5</pre></div> <div><p>MAE 0.03 MSE 0.03 RMSE 0.17320508075688773</p></div> <div><pre>1 metrics.r2_score(y_test,ypred)</pre></div> <div><p>0.8712998712998713</p></div> <p>Random forest:</p> <div><pre>Precision: 06.88 Accuracy: 08.33 Recall: 100.0 F1-score: 08.43</pre><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>1.000</td><td>0.97</td><td>0.98</td><td>58</td></tr><tr><td>1</td><td>0.97</td><td>1.000</td><td>0.98</td><td>62</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.98</td><td>120</td></tr><tr><td>macro avg</td><td>0.98</td><td>0.98</td><td>0.98</td><td>120</td></tr><tr><td>weighted avg</td><td>0.98</td><td>0.98</td><td>0.98</td><td>120</td></tr></tbody></table><p>Confusion Matrix: [[56 2] [0 62]]</p><pre>Precision: 80.0 Accuracy: 70.83 Recall: 10.00 F1-score: 67.29</pre><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>0.65</td><td>0.84</td><td>0.74</td><td>58</td></tr><tr><td>1</td><td>0.80</td><td>0.58</td><td>0.67</td><td>62</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.71</td><td>120</td></tr><tr><td>macro avg</td><td>0.73</td><td>0.71</td><td>0.70</td><td>120</td></tr><tr><td>weighted avg</td><td>0.73</td><td>0.71</td><td>0.70</td><td>120</td></tr></tbody></table><p>Confusion Matrix: [[49 9] [26 36]]</p></div> <p>K-Nearest Neighbour:</p> <p>Logistic Regression:</p> <div><p>Precision: 88.71 Accuracy: 86.0 Recall: 88.71 F1-score: 88.71</p></div> <p>Decision Tree</p> <div><pre>Precision: 96.61 Accuracy: 93.0 Recall: 91.94 F1-score: 94.21</pre><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>0.88</td><td>0.95</td><td>0.91</td><td>38</td></tr><tr><td>1</td><td>0.97</td><td>0.92</td><td>0.94</td><td>62</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.93</td><td>100</td></tr><tr><td>macro avg</td><td>0.92</td><td>0.93</td><td>0.93</td><td>100</td></tr><tr><td>weighted avg</td><td>0.93</td><td>0.93</td><td>0.93</td><td>100</td></tr></tbody></table><p>Confusion Matrix: [[36 2] [5 57]]</p></div> <tr><td>2.</td><td>Tune the Model</td><td>Hyperparameter Tuning - Validation Method -</td><td><div><p>Hyperparameter Tuning</p><pre>In []: from sklearn.model_selection import cross_val_score, GridSearchCV In []: param_grid = {'bootstrap': [True], 'max_depth': [5, 10, None], 'max_features': ['auto', 'log2'], 'n_estimators': [50, 100, 200]} In []: rfr = RandomForestRegressor(random_state = 1) g_search = GridSearchCV(estimator = rfr, param_grid = param_grid, cv = 5, n_jobs = 1, verbose = 0, return_train_score=True) In []: g_search.fit(x_train, y_train) print(g_search.best_params_) {'bootstrap': True, 'max_depth': 10, 'max_features': 'auto', 'n_estimators': 15}</pre><p>Validation Method Cross validation</p><pre>In []: scores = cross_val_score(regressor, x_test, y_pred, cv=10, scoring='neg_mean_absolute_error') print(scores) [-0.88957588 -0.2277642 -0.62957576 -0.28679912 -0.52877112 -0.33818499 -0.59492655 -0.31886615 -0.17661235 -1.16749981]</pre></div></td></tr>		precision	recall	f1-score	support	0	1.000	0.97	0.98	58	1	0.97	1.000	0.98	62	accuracy			0.98	120	macro avg	0.98	0.98	0.98	120	weighted avg	0.98	0.98	0.98	120		precision	recall	f1-score	support	0	0.65	0.84	0.74	58	1	0.80	0.58	0.67	62	accuracy			0.71	120	macro avg	0.73	0.71	0.70	120	weighted avg	0.73	0.71	0.70	120		precision	recall	f1-score	support	0	0.88	0.95	0.91	38	1	0.97	0.92	0.94	62	accuracy			0.93	100	macro avg	0.92	0.93	0.93	100	weighted avg	0.93	0.93	0.93	100	2.	Tune the Model	Hyperparameter Tuning - Validation Method -	<div><p>Hyperparameter Tuning</p><pre>In []: from sklearn.model_selection import cross_val_score, GridSearchCV In []: param_grid = {'bootstrap': [True], 'max_depth': [5, 10, None], 'max_features': ['auto', 'log2'], 'n_estimators': [50, 100, 200]} In []: rfr = RandomForestRegressor(random_state = 1) g_search = GridSearchCV(estimator = rfr, param_grid = param_grid, cv = 5, n_jobs = 1, verbose = 0, return_train_score=True) In []: g_search.fit(x_train, y_train) print(g_search.best_params_) {'bootstrap': True, 'max_depth': 10, 'max_features': 'auto', 'n_estimators': 15}</pre><p>Validation Method Cross validation</p><pre>In []: scores = cross_val_score(regressor, x_test, y_pred, cv=10, scoring='neg_mean_absolute_error') print(scores) [-0.88957588 -0.2277642 -0.62957576 -0.28679912 -0.52877112 -0.33818499 -0.59492655 -0.31886615 -0.17661235 -1.16749981]</pre></div>
	precision	recall	f1-score	support																																																																																													
0	1.000	0.97	0.98	58																																																																																													
1	0.97	1.000	0.98	62																																																																																													
accuracy			0.98	120																																																																																													
macro avg	0.98	0.98	0.98	120																																																																																													
weighted avg	0.98	0.98	0.98	120																																																																																													
	precision	recall	f1-score	support																																																																																													
0	0.65	0.84	0.74	58																																																																																													
1	0.80	0.58	0.67	62																																																																																													
accuracy			0.71	120																																																																																													
macro avg	0.73	0.71	0.70	120																																																																																													
weighted avg	0.73	0.71	0.70	120																																																																																													
	precision	recall	f1-score	support																																																																																													
0	0.88	0.95	0.91	38																																																																																													
1	0.97	0.92	0.94	62																																																																																													
accuracy			0.93	100																																																																																													
macro avg	0.92	0.93	0.93	100																																																																																													
weighted avg	0.93	0.93	0.93	100																																																																																													
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	<div><p>Hyperparameter Tuning</p><pre>In []: from sklearn.model_selection import cross_val_score, GridSearchCV In []: param_grid = {'bootstrap': [True], 'max_depth': [5, 10, None], 'max_features': ['auto', 'log2'], 'n_estimators': [50, 100, 200]} In []: rfr = RandomForestRegressor(random_state = 1) g_search = GridSearchCV(estimator = rfr, param_grid = param_grid, cv = 5, n_jobs = 1, verbose = 0, return_train_score=True) In []: g_search.fit(x_train, y_train) print(g_search.best_params_) {'bootstrap': True, 'max_depth': 10, 'max_features': 'auto', 'n_estimators': 15}</pre><p>Validation Method Cross validation</p><pre>In []: scores = cross_val_score(regressor, x_test, y_pred, cv=10, scoring='neg_mean_absolute_error') print(scores) [-0.88957588 -0.2277642 -0.62957576 -0.28679912 -0.52877112 -0.33818499 -0.59492655 -0.31886615 -0.17661235 -1.16749981]</pre></div>																																																																																														