

Assignment-Classification

Problem Statement:

To create a predictive model which will predict the Chronic Kidney Disease (CKD) based on the several parameters and from the given Client's Dataset.

Dataset:

The dataset has 27 columns and 399 rows. The data has patient related information to be used in all possible algorithms to train the model.

Input and Output classification:

Input - ['age', 'bp', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'pot', 'hrmo', 'pcv',
'wc', 'rc', 'sg_b', 'sg_c', 'sg_d', 'sg_e', 'rbc_normal', 'pc_normal',
'pcc_present', 'ba_present', 'htn_yes', 'dm_yes', 'cad_yes',
'appet_yes', 'pe_yes', 'ane_yes']

We must predict the Chronic Kidney Disease (CKD) for given input parameters. So, we took "classification_yes" as output parameters.

Output – ['classification_yes']

Pre-processing Method:

AI will not able handle the categorical data (as string) which contains in the part of input data. So, we must convert them into equal numeric data for further process. For that, we are used **get_dummies** method from Panda's library to convert them into numeric data.

Model accuracy test:

The Model accuracy test will be defined by several values such as **confusion_matrix**, **classification_report**, **f1_score** value, **roc_score**

The below machine learning algorithms are used to find the R2 Value .

Support Vector Machine:

Classification report:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	51
1	1.00	0.99	0.99	82
accuracy			0.99	133
macro avg	0.99	0.99	0.99	133
weighted avg	0.99	0.99	0.99	133

The f1_macro value for best parameter

{'C': 10, 'gamma': 'auto', 'kernel': 'sigmoid'}:

f1_score : 0.9924946382275899

roc_score : 1.0

Decision Tree:

Classification report:

	precision	recall	f1-score	support
0	0.96	0.98	0.97	51
1	0.99	0.98	0.98	82
accuracy			0.98	133
macro avg	0.97	0.98	0.98	133
weighted avg	0.98	0.98	0.98	133

The f1_macro value for best parameter

{'criterion': 'gini', 'max_features': 'log2', 'splitter': 'random'}

f1_score : 0.9774839146827697

roc_score : 0.978000956480153

Random Forest:

Classification report:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	51
1	1.00	0.99	0.99	82
accuracy			0.99	133
macro avg	0.99	0.99	0.99	133
weighted avg	0.99	0.99	0.99	133

The f1_macro value for best parameter

{'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 100}

f1_score : 0.9924946382275899

roc_score : 1.0

KNeighbors-KNN:

Classification report:

	precision	recall	f1-score	support
0	0.86	1.00	0.93	51
1	1.00	0.90	0.95	82
accuracy			0.94	133
macro avg	0.93	0.95	0.94	133
weighted avg	0.95	0.94	0.94	133

The f1_macro value for best parameter

{'algorithm': 'ball_tree', 'metric': 'minkowski', 'n_neighbors': 5}

f1_score : 0.9404945931261721

roc_score : 0.9992826398852224

Logistic-Regression:

Classification report:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	51
1	1.00	0.99	0.99	82
accuracy			0.99	133
macro avg	0.99	0.99	0.99	133
weighted avg	0.99	0.99	0.99	133

The f1_macro value for best parameter

{'penalty': 'l2', 'solver': 'newton-cg'}

f1_score : 0.9924946382275899

roc_score : 1.0

Final Decision:

We are finally decided to create a model with “SVM, Random Forest, Logistic-Regression” algorithm. It has 0.99% of accuracy and roc_score is 1.0 than other algorithms based few parameter values which we calculated with various algorithms.

Thank you !