

SVM_Chronic Kidney Disease Prediction

Solution:

1) Find Kidney Disease Positive or Negative. If any possible to get positive it will give suggestion

2) The dataset has 399 rows and 25 columns

3) **Best Model** gamma=auto, c=10, kernel=sigmoid

Accuracy 99%

Roc_value=1

F1_score=99%(0 and 1) as well as error1 has less than compare to error 2 so this model is learning good that's why I choose kernel=sigmoid predict model

```
:
from sklearn.metrics import f1_score
f1_macro=f1_score(y_test,grid_predictions,average='weighted')
print("The f1_macro value for best parameter {}".format(grid.best_params_),f1_macro)

The f1_macro value for best parameter {'C': 10, 'gamma': 'auto', 'kernel': 'sigmoid'}: 0.9924946382275899

:
print("The confusion Matrix:\n",cm)

The confusion Matrix:
[[51  0]
 [ 1 81]]

:
print("The report:\n",clf_report)

The 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
 macro avg          0.99
weighted avg          0.99
```

```
[42]: table=pd.DataFrame.from_dict(re)
      table
```

[42]:	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_C	param_gamma	param_kernel	params	split0_test_score	split1_test_score	split2_test_score
0	0.013260	0.010138	0.012748	0.006063	10	auto	linear	{'C': 10, 'gamma': 'auto', 'kernel': 'linear'}	0.981569	0.962636	0.962573
1	0.018338	0.011457	0.024069	0.002535	10	auto	rbf	{'C': 10, 'gamma': 'auto', 'kernel': 'rbf'}	0.963284	0.981014	1.000000
2	0.026197	0.012815	0.015524	0.014832	10	auto	poly	{'C': 10, 'gamma': 'auto', 'kernel': 'poly'}	1.000000	0.981014	1.000000
3	0.004356	0.001432	0.006214	0.004846	10	auto	sigmoid	{'C': 10, 'gamma': 'auto', 'kernel': 'sigmoid'}	0.981569	1.000000	1.000000
4	0.005450	0.004329	0.014302	0.005088	10	scale	linear	{'C': 10, 'gamma': 'scale', 'kernel': 'linear'}	0.981569	0.962636	0.962573
5	0.016542	0.003591	0.008925	0.005969	10	scale	rbf	{'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}	0.963284	0.981014	1.000000
6	0.026524	0.007157	0.018694	0.004184	10	scale	poly	{'C': 10, 'gamma': 'scale', 'kernel': 'poly'}	1.000000	0.981014	1.000000
7	0.016751	0.008767	0.021445	0.003313	10	scale	sigmoid	{'C': 10, 'gamma': 'scale', 'kernel': 'sigmoid'}	0.981569	1.000000	1.000000
8	0.015566	0.002036	0.015812	0.004327	100	auto	linear	{'C': 100, 'gamma': 'auto', 'kernel': 'linear'}	0.981569	0.962636	0.962573
9	0.011303	0.006902	0.021021	0.011483	100	auto	rbf	{'C': 100, 'gamma': 'auto', 'kernel': 'rbf'}	0.963284	0.981014	1.000000
10	0.027932	0.005216	0.018465	0.013473	100	auto	poly	{'C': 100, 'gamma': 'auto', 'kernel': 'poly'}	0.963284	0.981014	1.000000
11	0.015996	0.004191	0.015770	0.008849	100	auto	sigmoid	{'C': 100, 'gamma': 'auto', 'kernel': 'sigmoid'}	0.981569	0.943699	0.944023
12	0.014892	0.007726	0.004011	0.004912	100	scale	linear	{'C': 100, 'gamma': 'scale', 'kernel': 'linear'}	0.981569	0.962636	0.962573
13	0.021203	0.015213	0.024305	0.016257	100	scale	rbf	{'C': 100, 'gamma': 'scale', 'kernel': 'rbf'}	0.963284	0.981014	1.000000
14	0.032890	0.017884	0.019330	0.005756	100	scale	poly	{'C': 100, 'gamma': 'scale', 'kernel': 'poly'}	0.963284	0.981014	1.000000
15	0.021981	0.012704	0.019472	0.005009	100	scale	sigmoid	{'C': 100, 'gamma': 'scale', 'kernel': 'sigmoid'}	0.981569	0.943699	0.944023

16	0.011999	0.007499	0.016658	0.006810	1000	auto	linear	{'C': 1000, 'gamma': 'auto', 'kernel': 'linear'}	0.981569	0.962636	0.962573
17	0.022121	0.001980	0.023908	0.004313	1000	auto	rbf	{'C': 1000, 'gamma': 'auto', 'kernel': 'rbf'}	0.963284	0.981014	1.000000
18	0.032712	0.013283	0.023261	0.006490	1000	auto	poly	{'C': 1000, 'gamma': 'auto', 'kernel': 'poly'}	0.963284	0.981014	0.981217
19	0.020044	0.005166	0.022175	0.011070	1000	auto	sigmoid	{'C': 1000, 'gamma': 'auto', 'kernel': 'sigmoid'}	0.981569	0.943699	0.962573
20	0.017006	0.005980	0.020916	0.003228	1000	scale	linear	{'C': 1000, 'gamma': 'scale', 'kernel': 'linear'}	0.981569	0.962636	0.962573
21	0.025105	0.015836	0.021040	0.000760	1000	scale	rbf	{'C': 1000, 'gamma': 'scale', 'kernel': 'rbf'}	0.963284	0.981014	1.000000
22	0.038331	0.014897	0.013804	0.005718	1000	scale	poly	{'C': 1000, 'gamma': 'scale', 'kernel': 'poly'}	0.963284	0.981014	0.981217
23	0.017378	0.005918	0.013243	0.007517	1000	scale	sigmoid	{'C': 1000, 'gamma': 'scale', 'kernel': 'sigmoid'}	0.981569	0.943699	0.962573
24	0.016222	0.004898	0.016460	0.006742	2000	auto	linear	{'C': 2000, 'gamma': 'auto', 'kernel': 'linear'}	0.981569	0.962636	0.962573
25	0.014604	0.008478	0.014858	0.007409	2000	auto	rbf	{'C': 2000, 'gamma': 'auto', 'kernel': 'rbf'}	0.963284	0.981014	1.000000
26	0.024850	0.005280	0.012562	0.006069	2000	auto	poly	{'C': 2000, 'gamma': 'auto', 'kernel': 'poly'}	0.963284	0.981014	0.981217
27	0.018921	0.007214	0.010742	0.005162	2000	auto	sigmoid	{'C': 2000, 'gamma': 'auto', 'kernel': 'sigmoid'}	0.981569	0.943699	0.962573

28	0.017957	0.003528	0.008879	0.003188	2000	scale	linear	{ 'gamma': 'scale', 'kernel': 'linear'}	0.981569	0.962636	0.962573
29	0.011631	0.005182	0.021643	0.006995	2000	scale	rbf	{ 'C': 2000, 'gamma': 'scale', 'kernel': 'rbf'}	0.963284	0.981014	1.000000
30	0.030860	0.008593	0.015463	0.008767	2000	scale	poly	{ 'C': 2000, 'gamma': 'scale', 'kernel': 'poly'}	0.963284	0.981014	0.981217
31	0.023689	0.014605	0.005642	0.011285	2000	scale	sigmoid	{ 'C': 2000, 'gamma': 'scale', 'kernel': 'sigmo...	0.981569	0.943699	0.962573
32	0.016312	0.006068	0.028998	0.006256	3000	auto	linear	{ 'C': 3000, 'gamma': 'auto', 'kernel': 'linear'}	0.981569	0.962636	0.962573
33	0.024416	0.008191	0.021410	0.007342	3000	auto	rbf	{ 'C': 3000, 'gamma': 'auto', 'kernel': 'rbf'}	0.963284	0.981014	1.000000
34	0.031515	0.005735	0.024537	0.012504	3000	auto	poly	{ 'C': 3000, 'gamma': 'auto', 'kernel': 'poly'}	0.963284	0.981014	0.981217
35	0.023297	0.009361	0.018927	0.002699	3000	auto	sigmoid	{ 'C': 3000, 'gamma': 'auto', 'kernel': 'sigmoid'}	0.981569	0.943699	0.962573
36	0.007783	0.005998	0.020792	0.009686	3000	scale	linear	{ 'C': 3000, 'gamma': 'scale', 'kernel': 'linear'}	0.981569	0.962636	0.962573
37	0.015487	0.006156	0.029971	0.006642	3000	scale	rbf	{ 'C': 3000, 'gamma': 'scale', 'kernel': 'rbf'}	0.963284	0.981014	1.000000
38	0.035821	0.012198	0.025965	0.012189	3000	scale	poly	{ 'C': 3000, 'gamma': 'scale', 'kernel': 'poly'}	0.963284	0.981014	0.981217
39	0.014501	0.012191	0.011484	0.012956	3000	scale	sigmoid	{ 'C': 3000, 'gamma': 'scale', 'kernel': 'sigmo...	0.981569	0.943699	0.962573