Classification Assignment

1. Problem Statement

Predict the Chronic Kidney Disease (CKD) on the several parameters.

2. Dataset Information

Dataset contains 27 columns and 400 rows of data values including header

3. Data Preprocessing

In the given dataset, there are twelve columns contains string values. Since Python AI models cannot directly handle categorical data, so those column's values will be converted into meaningful numerical data.

4. R&D on Model Selection
    a. Logistics Regression

The Mean test score value obtained in the Logistics Regression model is **0.96**

Sl No	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_penalty	param_solver	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
0	0.013201	0.001621	0.002909	0.000971	l2	lbfgs	{'penalty': 'l2', 'solver': 'lbfgs'}	0.851852	0.942166	0.925146	0.886792	0.961826	0.913557	0.039513	4
1	0.005597	0.002373	0.001653	0.000728	l2	liblinear	{'penalty': 'l2', 'solver': 'liblinear'}	0.962963	0.981014	0.962573	0.943651	0.961826	0.962405	0.011821	3
2	0.033414	0.004342	0.002543	0.002102	l2	newton-cg	{'penalty': 'l2', 'solver': 'newton-cg'}	0.981569	0.981014	0.962573	0.962264	0.961826	0.969849	0.009347	1
3	0.004749	0.003465	0.001214	0.000188	l2	newton-cholesky	{'penalty': 'l2', 'solver': 'newton-cholesky'}	0.981569	0.981014	0.962573	0.962264	0.961826	0.969849	0.009347	1
4	0.006982	0.001705	0.001139	0.00002	l2	sag	{'penalty': 'l2', 'solver': 'sag'}	0.486532	0.5014	0.477841	0.477841	0.477841	0.484293	0.009196	5
5	0.006413	0.001674	0.001185	0.000137	l2	saga	{'penalty': 'l2', 'solver': 'saga'}	0.486532	0.5014	0.477841	0.477841	0.477841	0.484293	0.009196	5

b. K-Neighbors Classifier

The Mean test score value obtained in the Logistics Regression model is **0.75**

Sl No	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_algorithm	param_weights	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
0	0.003734	0.001524	0.004948	0.001698	auto	uniform	{'algorithm': 'auto', 'weights': 'uniform'}	0.744423	0.685491	0.664016	0.702707	0.677527	0.694833	0.027776	5
1	0.00317	0.000541	0.004565	0.002366	auto	distance	{'algorithm': 'auto', 'weights': 'distance'}	0.706269	0.685491	0.701348	0.758432	0.738011	0.71791	0.026483	1
2	0.00223	0.001132	0.003722	0.001195	ball_tree	uniform	{'algorithm': 'ball_tree', 'weights': 'uniform'}	0.744423	0.685491	0.664016	0.702707	0.677527	0.694833	0.027776	5
3	0.002457	0.000773	0.002859	0.000892	ball_tree	distance	{'algorithm': 'ball_tree', 'weights': 'distance'}	0.706269	0.685491	0.701348	0.758432	0.738011	0.71791	0.026483	1
4	0.001203	0.000063	0.002294	0.000052	kd_tree	uniform	{'algorithm': 'kd_tree', 'weights': 'uniform'}	0.744423	0.685491	0.664016	0.702707	0.677527	0.694833	0.027776	5
5	0.001372	0.000474	0.001644	0.000289	kd_tree	distance	{'algorithm': 'kd_tree', 'weights': 'distance'}	0.706269	0.685491	0.701348	0.758432	0.738011	0.71791	0.026483	1
6	0.001488	0.000376	0.002854	0.000568	brute	uniform	{'algorithm': 'brute', 'weights': 'uniform'}	0.744423	0.685491	0.664016	0.702707	0.677527	0.694833	0.027776	5
7	0.000952	0.000097	0.001455	0.000047	brute	distance	{'algorithm': 'brute', 'weights': 'distance'}	0.706269	0.685491	0.701348	0.758432	0.738011	0.71791	0.026483	1

c. Naive Bayes – GaussianNB

The GaussianNB model is **0.98**

	precision	recall	f1-score	support
0	0.94	1.00	0.97	51
1	1.00	0.96	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

d. Naive Bayes – MultinomialNB

The MultinomialNB model is **0.83**

	precision	recall	f1-score	support
0	0.69	0.98	0.81	51
1	0.98	0.73	0.84	82
accuracy			0.83	133
macro avg	0.84	0.86	0.83	133
weighted avg	0.87	0.83	0.83	133

e. Naive Bayes – ComplementNB

The ComplementNB model is **0.95**

	precision	recall	f1-score	support
0	0.89	1.00	0.94	51
1	1.00	0.93	0.96	82
accuracy			0.95	133
macro avg	0.95	0.96	0.95	133
weighted avg	0.96	0.95	0.96	133

f. Random Forest

The Mean test score value obtained in the Logistics Regression model is **0.98**

('class_weight': 'balanced', 'criterion': 'gini', 'max_features': 'log2')

Sl No	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_class_weight	param_criterion	param_max_features	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
0	0.00111	0.00013	0	0	None	gini	sqrt	{'class_weight': 'None', 'criterion': 'gini', ...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	13
1	0.00103	4.6E-05	0	0	None	gini	log2	{'class_weight': 'None', 'criterion': 'gini', ...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	13
2	0.00136	0.00081	0	0	None	entropy	sqrt	{'class_weight': 'None', 'criterion': 'entropy...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	13
3	0.00081	1.8E-05	0	0	None	entropy	log2	{'class_weight': 'None', 'criterion': 'entropy...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	13
4	0.00095	0.00015	0	0	None	log_loss	sqrt	{'class_weight': 'None', 'criterion': 'log_los...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	13
5	0.00399	0.00379	0	0	None	log_loss	log2	{'class_weight': 'None', 'criterion': 'log_los...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	13
6	0.18961	0.02127	0.01077	0.00657	balanced	gini	sqrt	{'class_weight': 'balanced', 'criterion': 'gin...	0.98138	0.96176	0.98122	0.96226	0.98103	0.97353	0.00941	11
7	0.15735	0.02623	0.01112	0.0053	balanced	gini	log2	{'class_weight': 'balanced', 'criterion': 'gin...	1	0.98101	0.98122	0.98103	1	0.98865	0.00927	1
8	0.14445	0.01607	0.0066	0.00289	balanced	entropy	sqrt	{'class_weight': 'balanced', 'criterion': 'ent...	0.96296	0.96176	0.98122	0.96226	1	0.97364	0.01508	10
9	0.17056	0.0095	0.00901	0.00275	balanced	entropy	log2	{'class_weight': 'balanced', 'criterion': 'ent...	1	0.98101	0.98122	0.96226	1	0.9849	0.01412	4
10	0.20649	0.07015	0.00841	0.00231	balanced	log_loss	sqrt	{'class_weight': 'balanced', 'criterion': 'log...	0.96296	0.98101	0.98122	0.96226	1	0.97749	0.01397	8
11	0.19106	0.05033	0.00838	0.00515	balanced	log_loss	log2	{'class_weight': 'balanced', 'criterion': 'log...	1	0.98101	0.98122	0.98103	1	0.98865	0.00927	1
12	0.2075	0.03173	0.00646	0.00336	balanced_s	gini	sqrt	{'class_weight': 'balanced_subsample', 'criter...	0.98138	0.98101	0.98122	0.98103	1	0.98493	0.00754	3
13	0.24024	0.01989	0.00861	0.00507	balanced_s	gini	log2	{'class_weight': 'balanced_subsample', 'criter...	1	0.98101	0.98122	0.98103	0.98103	0.98486	0.00757	5
14	0.22418	0.02325	0.00753	0.00516	balanced_s	entropy	sqrt	{'class_weight': 'balanced_subsample', 'criter...	0.96296	0.98101	0.98122	0.96226	1	0.97749	0.01397	8
15	0.28915	0.07733	0.01403	0.0069	balanced_s	entropy	log2	{'class_weight': 'balanced_subsample', 'criter...	0.98138	0.98101	0.98122	0.96226	1	0.98118	0.01193	6
16	0.27892	0.01557	0.00954	0.00321	balanced_s	log_loss	sqrt	{'class_weight': 'balanced_subsample', 'criter...	0.98138	0.94217	0.98122	0.96226	0.98103	0.96961	0.01556	12
17	0.22345	0.02496	0.01034	0.00111	balanced_s	log_loss	log2	{'class_weight': 'balanced_subsample', 'criter...	0.96328	0.98101	0.98122	0.96226	1	0.97756	0.0139	7

g. Random Forest

The Mean test score value obtained in the Logistics Regression model is **0.95**

('criterion': 'gini', 'max_features': 'sqrt', 'splitter': 'random')

SL No	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_criterion	param_max_features	param_splitter	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
0	0.0587	0.0205	0.016717	0.006859	gini	sqrt	best	{'criterion': 'gini', 'max_features': 'sqrt', ...	0.94471	0.981014	0.924528	0.962264	0.981031	0.958709	0.021784	6
1	0.0301	0.0207	0.021218	0.016879	gini	sqrt	random	{'criterion': 'gini', 'max_features': 'sqrt', ...	1	1	0.962573	0.981031	0.981031	0.984927	0.014032	1
2	0.0069	0.0038	0.003922	0.000934	gini	log2	best	{'criterion': 'gini', 'max_features': 'log2', ...	0.9451	0.981014	0.832761	0.981031	0.981217	0.944225	0.057448	10
3	0.0035	0.0005	0.003419	0.000678	gini	log2	random	{'criterion': 'gini', 'max_features': 'log2', ...	1	0.981014	0.925524	0.962264	0.962573	0.966275	0.024678	5
4	0.0033	0.0002	0.003099	0.0005	entropy	sqrt	best	{'criterion': 'entropy', 'max_features': 'sqrt...	0.92657	0.92351	0.887719	0.943093	0.961826	0.928543	0.024557	12
5	0.003	0.0004	0.003083	0.000548	entropy	sqrt	random	{'criterion': 'entropy', 'max_features': 'sqrt...	0.98157	0.943699	0.981217	1	1	0.981297	0.020558	2
6	0.0027	9E-05	0.002763	0.000144	entropy	log2	best	{'criterion': 'entropy', 'max_features': 'log2...	0.98138	0.885265	0.907035	1	0.961826	0.947101	0.043861	9
7	0.003	0.0005	0.003275	0.000987	entropy	log2	random	{'criterion': 'entropy', 'max_features': 'log2...	0.96328	0.962264	0.981217	0.981217	0.981217	0.97384	0.009041	3
8	0.004	0.0013	0.002946	0.000326	log_loss	sqrt	best	{'criterion': 'log_loss', 'max_features': 'sqr...	0.94471	0.961755	0.944023	0.962264	0.943093	0.951169	0.008868	8
9	0.0053	0.0016	0.005112	0.001687	log_loss	sqrt	random	{'criterion': 'log_loss', 'max_features': 'sqr...	1	0.924528	0.925524	0.981031	0.943093	0.954835	0.030468	7
10	0.0029	0.0005	0.00293	0.000267	log_loss	log2	best	{'criterion': 'log_loss', 'max_features': 'log...	0.92657	0.962264	0.962573	0.906085	0.962264	0.943951	0.023467	11
11	0.0025	0.0002	0.002837	0.000495	log_loss	log2	random	{'criterion': 'log_loss', 'max_features': 'log...	1	0.922185	0.981217	0.962264	0.981217	0.969377	0.026442	4

In the above experiment, based on the accuracy score, the **Random Forest** (*'class_weight': 'balanced', 'criterion': 'gini', 'max_features': 'log2'*) model appears to be the best-performing model.