Assignment-Classification Algorithm

1.) Identify your problem statement

Want to predict the CKD Affected or not

2.) Tell basic info about the dataset

Total number of rows 399. Total number of columns 26

3.) Mention the pre-processing method

Yes, I do converting string to number-nominal data

ORIGINAL DATA

```
In [26]: 1 dataset
```

Out[26]:

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wc	rc	htn	dm	cad	appet	pe
0	2.000000	76.459948	c	3.0	0.0	normal	abnormal	notpresent	notpresent	148.112676	...	38.868902	8408.191126	4.705597	no	no	no	yes	yes
1	3.000000	76.459948	c	2.0	0.0	normal	normal	notpresent	notpresent	148.112676	...	34.000000	12300.000000	4.705597	no	no	no	yes	poor
2	4.000000	76.459948	a	1.0	0.0	normal	normal	notpresent	notpresent	99.000000	...	34.000000	8408.191126	4.705597	no	no	no	yes	poor
3	5.000000	76.459948	d	1.0	0.0	normal	normal	notpresent	notpresent	148.112676	...	38.868902	8408.191126	4.705597	no	no	no	yes	poor
4	5.000000	50.000000	c	0.0	0.0	normal	normal	notpresent	notpresent	148.112676	...	36.000000	12400.000000	4.705597	no	no	no	yes	poor
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
394	51.492308	70.000000	a	0.0	0.0	normal	normal	notpresent	notpresent	219.000000	...	37.000000	9800.000000	4.400000	no	no	no	yes	poor
395	51.492308	70.000000	c	0.0	2.0	normal	normal	notpresent	notpresent	220.000000	...	27.000000	8408.191126	4.705597	yes	yes	no	yes	poor
396	51.492308	70.000000	c	3.0	0.0	normal	normal	notpresent	notpresent	110.000000	...	26.000000	9200.000000	3.400000	yes	yes	no	poor	poor
397	51.492308	90.000000	a	0.0	0.0	normal	normal	notpresent	notpresent	207.000000	...	38.868902	8408.191126	4.705597	yes	yes	no	yes	poor
398	51.492308	80.000000	a	0.0	0.0	normal	normal	notpresent	notpresent	100.000000	...	53.000000	8500.000000	4.900000	no	no	no	yes	poor

399 rows × 25 columns

CONVERTING DATA

```
In [27]: 1 dataset=pd.get_dummies(dataset,drop_first=True)
```

```
In [28]: 1 dataset
```

Out[28]:

	age	bp	al	su	bgr	bu	sc	sod	pot	hrmo	...	pc_normal	pcc_present	ba_present	htn_yes	dm_
0	2.000000	76.459948	3.0	0.0	148.112676	57.482105	3.077356	137.528754	4.627244	12.518156	...	0	0	0	0	
1	3.000000	76.459948	2.0	0.0	148.112676	22.000000	0.700000	137.528754	4.627244	10.700000	...	1	0	0	0	
2	4.000000	76.459948	1.0	0.0	99.000000	23.000000	0.600000	138.000000	4.400000	12.000000	...	1	0	0	0	
3	5.000000	76.459948	1.0	0.0	148.112676	16.000000	0.700000	138.000000	3.200000	8.100000	...	1	0	0	0	
4	5.000000	50.000000	0.0	0.0	148.112676	25.000000	0.600000	137.528754	4.627244	11.800000	...	1	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
394	51.492308	70.000000	0.0	0.0	219.000000	36.000000	1.300000	139.000000	3.700000	12.500000	...	1	0	0	0	
395	51.492308	70.000000	0.0	2.0	220.000000	68.000000	2.800000	137.528754	4.627244	8.700000	...	1	0	0	1	
396	51.492308	70.000000	3.0	0.0	110.000000	115.000000	6.000000	134.000000	2.700000	9.100000	...	1	0	0	1	
397	51.492308	90.000000	0.0	0.0	207.000000	80.000000	6.800000	142.000000	5.500000	8.500000	...	1	0	0	1	
398	51.492308	80.000000	0.0	0.0	100.000000	49.000000	1.000000	140.000000	5.000000	16.300000	...	1	0	0	0	

399 rows × 28 columns

```
In [29]: 1 dataset=dataset.drop('age',axis=1)
```

```
In [30]: 1
         2 dataset
```

4.) Finally model

The **Random Forest** use Accuracy =0.99

5.) All research value of r2_score

1. SUPPORT VECTOR MACHINE:

	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 **SVM Regression** use Accuracy =0.99

2. DECISION TREE

	precision	recall	f1-score	support
0	0.96	1.00	0.98	45
1	1.00	0.97	0.99	75
accuracy			0.98	120
macro avg	0.98	0.99	0.98	120
weighted avg	0.98	0.98	0.98	120

The **DECISION TREE** use Accuracy =0.98

3. RANDOM FOREST

	precision	recall	f1-score	support
0	0.98	1.00	0.99	45
1	1.00	0.99	0.99	75
accuracy			0.99	120
macro avg	0.99	0.99	0.99	120
weighted avg	0.99	0.99	0.99	120

The **RANDOM FOREST** use Accuracy =0.99

4. Logistic Algorithm

	precision	recall	f1-score	support
0	0.66	1.00	0.79	79
1	0.00	0.00	0.00	41
accuracy			0.66	120
macro avg	0.33	0.50	0.40	120
weighted avg	0.43	0.66	0.52	120

The **Logistic Algorithm** use Accuracy =0.66

5. KNN classification

	precision	recall	f1-score	support
0	0.88	1.00	0.94	45
1	1.00	0.92	0.96	75
accuracy			0.95	120
macro avg	0.94	0.96	0.95	120
weighted avg	0.96	0.95	0.95	120

The **KNN classification** use Accuracy =0.66

6.) FINAL MODEL

The final model is Random Forest because the Accuracy value is high comparatively

Other value