

### Problem Statement or Requirement:

1.) Identify your problem statement:

AI USED AT THE PREDICT THE CHRONIC KIDNEY DISEASE

2.)basic info about the dataset (Total number of rows, columns):

399 rows x 25 columns

3.)Mention the pre-processing method if you're doing any (like converting string to number – nominal data)

pc_normal	pcc_present	ba_present	htn_yes	dm_yes	cad_yes	appet_yes	pe_yes	ane_yes	classification_yes
0	0	0	0	0	0	1	1	0	1
1	0	0	0	0	0	1	0	0	1
1	0	0	0	0	0	1	0	0	1
1	0	0	0	0	0	1	0	1	1
1	0	0	0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...
1	0	0	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0	1	1
1	0	0	1	1	0	0	0	0	1
1	0	0	1	1	0	1	0	1	1
1	0	0	0	0	0	1	0	0	0

4.)Develop a good model with good evaluation metric ,you have to come up with final model

1.Random forest CM = 0.9833333333333335

2.Support vector Machine CM = 0.9834018801410106

3.DECESION TREE CM= 0.9751481237656352

4..LOGISTIC CM = 0.9916844900066377

5.KNN CM= 0.9505208333333334

6. Naive Bayes CM= 0.9834018801410106

5.) All the research values (CM of the models) should be documented. (You can make tabulation or screenshot of the results.)

1.Random forest CM = 0.9833333333333335

```
The f1 macro value the best parameter{'class_weight': 'balanced', 'criterion': 'gini', 'max_features': 'log2', 'n_estimators': 100}: 0.9833333333333335
```

```
print(cm)
[[44  1]
 [ 1 74]]
```

```
print(clf_report )
```

	precision	recall	f1-score	support
0	0.98	0.98	0.98	48
1	0.99	0.99	0.99	75
accuracy			0.98	128
macro avg	0.98	0.98	0.98	128
weighted avg	0.98	0.98	0.98	128

2.Support vector Machine CM = 0.9834018801410106

```
The f1 macro value the best parameter{'C': 10, 'gamma': 'auto', 'kernel': 'sigmoid'}: 0.9834018801410106
```

```
print(cm)
[[45  0]
 [ 2 73]]
```

```
print(clf_report )
```

	precision	recall	f1-score	support
0	0.96	1.00	0.98	45
1	1.00	0.97	0.99	73
accuracy			0.98	120
macro avg	0.98	0.99	0.98	120
weighted avg	0.98	0.98	0.98	120

3.DECESION TREE CM= 0.9751481237656352

```
The f1 macro value the best parameter{'criterion': 'gini', 'max_features': 'log2', 'splitter': 'random'}: 0.9751481237656352
```

```
print(cm)
```

```
[[45  0]
 [ 3 72]]
```

```
print(clf_report )
```

	precision	recall	f1-score	support
0	0.98	1.00	0.99	45
1	0.99	0.97	0.98	72

0	0.94	1.00	0.97	45
1	1.00	0.96	0.98	75
accuracy			0.97	120
macro avg	0.97	0.98	0.97	120
weighted avg	0.98	0.97	0.98	120

4.LOGISTIC CM = 0.9916844900066377

The f1 macro value the best parameter{'penalty': 'l2', 'random\_state': 0, 'solver': 'lbfgs'}: 0.9916844900066377

```
print(cm)
```

```
[[45  0]
 [ 1 74]]
```

```
print(clf_report)
```

	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

5.KNN CM= 0.9505208333333334

The f1 macro value the best parameter{'algorithm': 'auto', 'n\_neighbors': 5, 'weights': 'distance'}: 0.9505208333333334

```
print(cm)
```

```
[[45  0]
 [ 6 69]]
```

```
print(clf_report )
```

	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

6. Naive Bayes CM= 0.9834018801410106

The f1 macro value the best parameter{'var\_smoothing': 0.006579332246575682}: 0.9834018801410106

```
[[45  0]
 [ 2 73]]
```

	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

6.) Mention your final model, justify why u have chosen the same.

1.LOGISTIC CM = 0.9916844900066377

2.CM-VALUE HIGH BUT COMPARISION ANOTHER MODEL