Classification Assignment

Problem Statement:

A requirement from the Hospital, Management asked us to create a predictive model which will predict the Chronic Kidney Disease (CKD) based on the several parameters. The Client has provided the dataset of the same.

1.) Identify your problem statement

To predict the Disease

a)Machine Learning

b)Supervised Learning

c)Classification

2) Tell basic info about the dataset

399 Rows × 25 columns

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

normal	abnormal	notpresent
yes	no	

These above terms are converted into numerical form

4.) Develop a good model with good evaluation metric. You can use any machine learning algorithm; you can create many models. Finally, you have to come up with final model.

1.LOGISTIC REGRESSION

```
print("The confusion matrix:\n",cm)
```

```
The confusion matrix:
 [[49  2]
 [ 0 82]]
```

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

```
The report:
              precision    recall  f1-score   support

           0       1.00      0.96      0.98        51
           1       0.98      1.00      0.99        82

    accuracy                           0.98       133
   macro avg       0.99      0.98      0.98       133
weighted avg       0.99      0.98      0.98       133
```

n_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
ewton-cg	{'penalty': 'l2', 'solver': 'newton-cg'}	0.945100	0.961755	0.981217	0.943093	0.981031	0.962439	0.016575	3
lbfgs	{'penalty': 'l2', 'solver': 'lbfgs'}	0.945100	0.981014	0.981217	0.943093	0.981031	0.966291	0.018133	1
liblinear	{'penalty': 'l2', 'solver': 'liblinear'}	0.945100	0.981014	0.981217	0.943093	0.981031	0.966291	0.018133	1
saga	{'penalty': 'l2', 'solver': 'saga'}	0.890759	0.868632	0.887719	0.832069	0.829279	0.861692	0.026452	4

2.KNN

```
print("The confusion matrix:\n",cm)
```

```
The confusion matrix:
 [[49  2]
 [17 65]]
```

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

```
The report:
              precision    recall  f1-score   support

           0       0.74      0.96      0.84        51
           1       0.97      0.79      0.87        82

    accuracy                           0.86       133
   macro avg       0.86      0.88      0.86       133
weighted avg       0.88      0.86      0.86       133
```

hts	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
rm	{'metric': 'euclidean', 'n_neighbors': 5, 'p':...	0.835979	0.831098	0.869925	0.813811	0.907035	0.851570	0.033170	5
nce	{'metric': 'euclidean', 'n_neighbors': 5, 'p':...	0.854345	0.831098	0.869925	0.813811	0.907035	0.855243	0.032244	3
rm	{'metric': 'manhattan', 'n_neighbors': 5, 'p':...	0.908877	0.887907	0.925524	0.869387	0.925146	0.903368	0.021860	1
nce	{'metric': 'manhattan', 'n_neighbors': 5, 'p':...	0.908877	0.887907	0.925524	0.850292	0.925146	0.899549	0.028209	2
rm	{'metric': 'minkowski', 'n_neighbors': 5, 'p':...	0.835979	0.831098	0.869925	0.813811	0.907035	0.851570	0.033170	5

3.RANDOM FOREST

```
The confusion matrix:
 [[51  0]
 [ 1 81]]
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       133
   macro avg       0.99      0.99      0.99       133
weighted avg       0.99      0.99      0.99       133
```

{'criterion': 'gini', 'max_features': 'sqrt', ...	1.000000	0.942166	0.981217	0.981031	0.981031	0.977089	0.018935	8
{'criterion': 'gini', 'max_features': 'log2', ...	0.981569	0.942166	0.981217	0.981031	0.943093	0.965815	0.018934	11
{'criterion': 'gini', 'max_features': 'log2', ...	1.000000	0.961755	0.981217	0.981031	0.981217	0.981044	0.012095	1
{'criterion': 'entropy', 'max_features': 'auto...	1.000000	0.961755	0.981217	0.981031	0.981031	0.981007	0.012095	2
{'criterion': 'entropy', 'max_features': 'auto...	1.000000	0.961755	0.981217	0.981031	0.981031	0.981007	0.012095	2

4.DECISION TREE

```
The confusion matrix:
 [[51  0]
 [ 1 81]]
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       133
   macro avg       0.99      0.99      0.99       133
weighted avg       0.99      0.99      0.99       133
```

om	{'criterion': 'entropy', 'max_features': 'auto...	0.908501	0.924528	0.906705	0.943651	0.943651	0.925407	0.016137	11
est	{'criterion': 'entropy', 'max_features': 'sqrt...	0.926567	0.962636	0.962573	0.903887	0.981031	0.947339	0.028003	3
om	{'criterion': 'entropy', 'max_features': 'sqrt...	0.908877	0.903610	0.944023	0.922492	0.962264	0.928253	0.022006	10
est	{'criterion': 'entropy', 'max_features': 'log2...	0.926567	0.962264	0.944023	0.923652	0.943651	0.940032	0.013949	5
om	{'criterion': 'entropy', 'max_features': 'log2...	0.945100	0.923510	0.962573	0.962264	0.962573	0.951204	0.015395	1

5.SUPPORT VECTOR MACHINE

```
The confusion matrix:
 [[51  0]
 [ 6 76]]
The report:
              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
```

sigmoid	'auto', 'kernel': 'sigmoid'}	0.890759	0.962264	0.962573	1.000000	0.962264	0.955572	0.035534	8
poly	{'C': 10, 'gamma': 'auto', 'kernel': 'poly'}	0.926978	0.981014	0.944023	1.000000	1.000000	0.970403	0.029820	1
linear	{'C': 10, 'gamma': 'scale', 'kernel': 'linear'}	0.890759	0.924528	0.981217	0.961826	0.962264	0.944119	0.032404	17
rbf	{'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}	0.908877	0.981014	0.944023	0.981217	1.000000	0.963026	0.032603	3

AT THE FINAL MODEL:

RANDOM FOREST

Because RF has more accuracy compare to the other algorithum

```
The confusion matrix:
 [[51  0]
 [ 1 81]]
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       133
   macro avg       0.99      0.99      0.99       133
weighted avg       0.99      0.99      0.99       133
```

THE OVERALL PERFORMANCE IS 0.99

0→not caused by disease

1→caused by disease

total count of not caused by disease =51

total count of caused by disease=82

{'criterion': 'gini', 'max_features': 'sqrt', ...	1.000000	0.942166	0.981217	0.981031	0.981031	0.977089	0.018935	8
{'criterion': 'gini', 'max_features': 'log2', ...	0.981569	0.942166	0.981217	0.981031	0.943093	0.965815	0.018934	11
{'criterion': 'gini', 'max_features': 'log2', ...	1.000000	0.961755	0.981217	0.981031	0.981217	0.981044	0.012095	1
{'criterion': 'entropy', 'max_features': 'auto...	1.000000	0.961755	0.981217	0.981031	0.981031	0.981007	0.012095	2
{'criterion': 'entropy', 'max_features': 'auto...	1.000000	0.961755	0.981217	0.981031	0.981031	0.981007	0.012095	2

By using the parameters are

n_estimators=100,

'criterion': 'gini',

'max_features': 'log2'

In these way we have to achieve the good accuracy