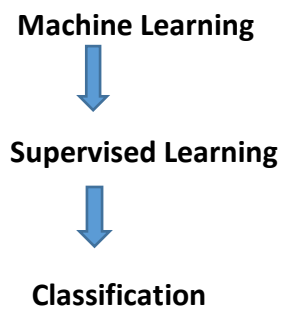


1.) Identify your problem statement :



2.) The dataset has the following basic information:

- Total number of rows: 398
- Total number of columns: 25

The columns in the dataset are:

1. age
2. bp
3. al
4. su
5. bgr
6. bu
7. sc
8. sod
9. pot
10. hrmo
11. pcv
12. wc
13. rc
14. sg
15. rbc
16. pc
17. pcc
18. ba
19. htn
20. dm
21. cad

- 22. appet
- 23. pe
- 24. ane
- 25. classification

3.) Mention the pre-processing method :-

Handle Categorical Variables:

Use one-hot encoding to convert **sg, rbc, pcc, pc, ba, htn, dm, cad, appet, pe, ane, classification** columns into numerical format.

Feature Scaling:

- Standardize numerical features.

Splitting the Data:

- Split the dataset into training and testing sets (1/3 testing).

4.) All the research values:

Machine Learning-Classification-Confusion Matrix

Dataset: CKD

1. RandomForest_Grid_Classifier:

Confusion Matrix value for best parameter: {'criterion': 'gini', 'max_features': 'sqrt', 'n_estimators': 10}

```
print(cm)
print(clf_report)
```

```
[[50  1]
 [ 1 81]]
```

	precision	recall	f1-score	support
0	0.98	0.98	0.98	51
1	0.99	0.99	0.99	82
accuracy			0.98	133
macro avg	0.98	0.98	0.98	133
weighted avg	0.98	0.98	0.98	133

Classification Report **accuracy = 0.98**

2.DecisionTree_Grid_Classifier:

Confusion Matrix value for best parameter: {'criterion': 'gini', 'max_features': 'log2', 'splitter': 'random'}

```
print(cm)
print(clf_report)
```

[[50 1]					
[2 80]]					
	precision	recall	f1-score	support	
0	0.96	0.98	0.97	51	
1	0.99	0.98	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	

Classification Report **accuracy = 0.98**

3.SVM_Grid_Classifier:

Confusion Matrix value for best parameter: {'C': 10, 'gamma': 'scale', 'kernel': 'sigmoid'}

```
print(cm)
print(clf_report)
```

[[51 0]					
[1 81]]					
	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	

Classification Report **accuracy = 0.99**

4.LogisticRegression_Grid_Classifier:

Confusion Matrix value for best parameter: {'penalty': 'l2', 'solver': 'sag'}

```
print(cm)
print(clf_report)
```

```
[[51  0]
 [ 1 81]]
```

	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

Classification Report **accuracy = 0.99**

5.KNN_Grid_Classifier:

Confusion Matrix value for best parameter: {'algorithm': 'auto', 'metric': 'minkowski', 'n_neighbors': 5, 'weights': 'distance'}

```
print(cm)
print(clf_report)
```

```
[[51  0]
 [ 8 74]]
```

	precision	recall	f1-score	support
0	0.86	1.00	0.93	51
1	1.00	0.90	0.95	82
accuracy			0.94	133
macro avg	0.93	0.95	0.94	133
weighted avg	0.95	0.94	0.94	133

Classification Report **accuracy = 0.94**

6.1.Naive_Baye's_Classifier_MultinomialNB:

Confusion Matrix value for best parameter: {'alpha': 1, 'class_prior': None, 'fit_prior': True, 'force_alpha': True}

[[50 1] [23 59]]					
		precision	recall	f1-score	support
	0	0.68	0.98	0.81	51
	1	0.98	0.72	0.83	82
accuracy				0.82	133
macro avg		0.83	0.85	0.82	133
weighted avg		0.87	0.82	0.82	133

Classification Report **accuracy = 0.82**

6.2.Naive_Baye's_Classifier_ BernoulliNB:

Confusion Matrix value for best parameter: {'alpha': 1, 'binarize': 0.0, 'class_prior': None, 'fit_prior': True, 'force_alpha': True}

[[51 0] [3 79]]					
		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

Classification Report **accuracy = 0.98**

6.3.Naive_Baye's_Classifier_ ComplementNB:

Confusion Matrix value for best parameter: {'alpha': 0.1, 'fit_prior': True, 'force_alpha': True, 'norm': False}

[[50 1] [20 62]]					
		precision	recall	f1-score	support
	0	0.71	0.98	0.83	51
	1	0.98	0.76	0.86	82
accuracy				0.84	133
macro avg		0.85	0.87	0.84	133
weighted avg		0.88	0.84	0.84	133

Classification Report **accuracy = 0.84**

6.4.Naive_Baye's_Classifier_ GaussianNB:

Confusion Matrix value for best parameter: {'priors': None, 'var_smoothing': 0}

```
print(cm)
print(clf_report)
```

```
[[51  0]
 [ 3 79]]
```

	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

Classification Report **accuracy = 0.98**

6.5.Naive_Baye's_Classifier CategoricalNB:

```
print(cm)
print(clf_report)
```

Confusion Matrix value for best parameter: {'alpha': 0.1, 'fit_prior': True, 'force_alpha': True}

```
[[150  0]
 [ 1 248]]
```

	precision	recall	f1-score	support
0	0.99	1.00	1.00	150
1	1.00	1.00	1.00	249
accuracy			1.00	399
macro avg	1.00	1.00	1.00	399
weighted avg	1.00	1.00	1.00	399

Classification Report **accuracy = 1.00**

5.) The Final Machine Learning best method of Classification is

LogisticRegression_Grid_Classification

Confusion Matrix value for best parameter: {'penalty': 'l2', 'solver': 'sag'}

```
print(cm)
print(clf_report)
```

```
[[51  0]
 [ 1 81]]
```

	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

Classification Report **accuracy = 0.99**