Chronic Kidney Disease Prediction

Problem Statement:

- **Machine Learning →** Here the input is in excel format, so we can consider the **Input as Numbers.** So in AI if input is Number we can give solution with Machine Learning.
- **Supervised →** In this problem we have clear idea about the requirement and the dataset, so we can continue with Supervised learning.
- **Classification →** Here the output is in categorical type (Classification), so we can continue with Classification.

Machine Learning → Supervised Learning → Classification

Basic Information about dataset:

- It has 25 columns and 339 rows.
- This dataset includes the client's personal details as input like Age, BP, Sugar etc.
- We need to predict the output as Classification (CKD is yes/no) and the output details are given in the dataset.

Pre-processing Method:

- In this model Pre-processing method is done for 12 columns which is

  - **Ordinal data**
    1. RBC (normal/abnormal) to (0/1)
    2. PC (normal/abnormal) to (0/1)
    3. PCC (present/notpresent) to (0/1)
    4. BA (present/notpresent) to (0/1)
    5. HTN (yes/no) to (0/1)
    6. DM (yes/no) to (0/1)
    7. CAD (yes/no) to (0/1)
    8. Appet (yes/poor) to (0/1)
    9. PE (yes/poor) to (0/1)
    10. ANE (yes/no) to (0/1)
    11. Classification (yes/no) to (0/1)

  - **Nominal data**
    1. SG (a/b/c/d/e) to (0000/1000/0100/0010/0001)

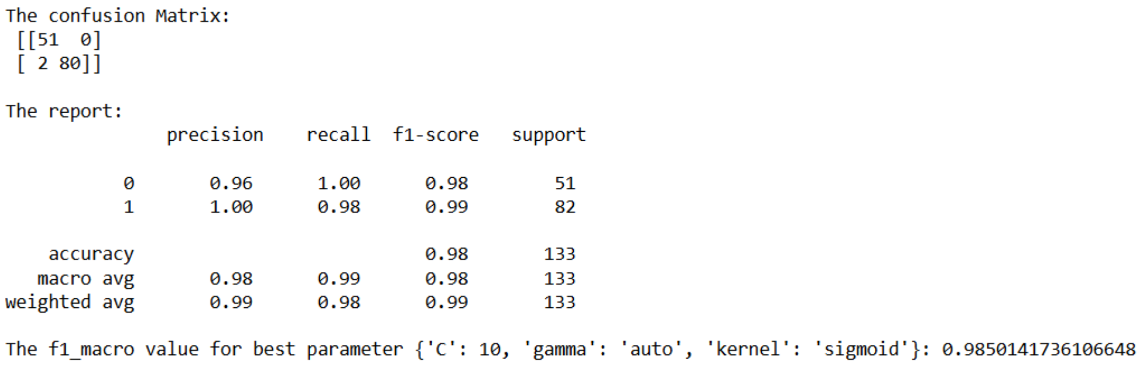
Machine Learning Algorithms:

1. Support Vector Machine
2. Decision Tree
3. Random Forest
4. Logistic Regression
5. KNN
6. Navie Bayes

1. Support Vector Machine:

Accuracy = 0.98

ROC_AUC = 1.0

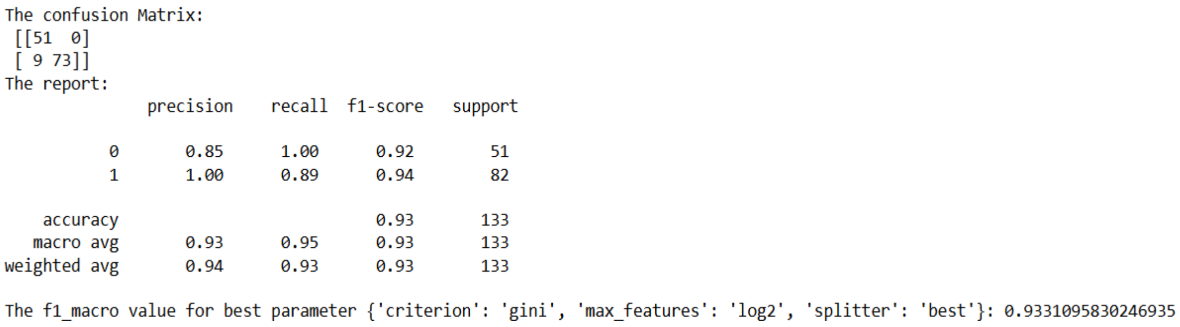


```
The confusion Matrix:
 [[51  0]
 [ 2 80]]

The report:
               precision    recall  f1-score   support

           0       0.96      1.00      0.98        51
           1       1.00      0.98      0.99        82

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

The f1_macro value for best parameter {'C': 10, 'gamma': 'auto', 'kernel': 'sigmoid'}: 0.9850141736106648
```

2. Decision Tree:

Accuracy = 0.93

ROC_AUC = 0.94

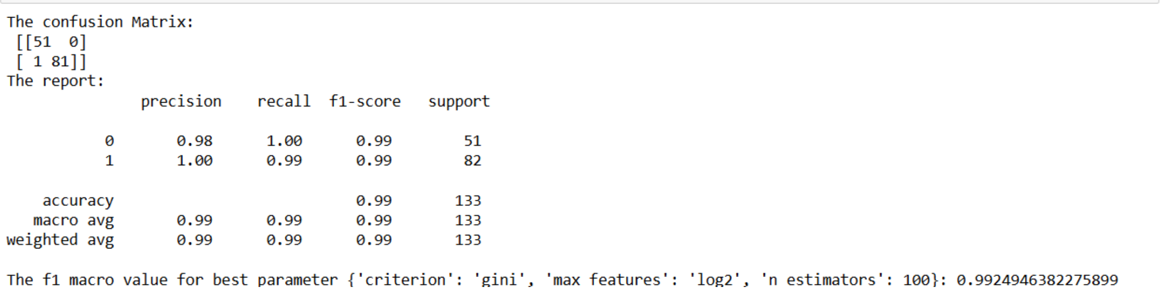


```
The confusion Matrix:
 [[51  0]
 [ 9 73]]
The report:
               precision    recall  f1-score   support

           0       0.85      1.00      0.92        51
           1       1.00      0.89      0.94        82

    accuracy                           0.93       133
   macro avg       0.93      0.95      0.93       133
weighted avg       0.94      0.93      0.93       133

The f1_macro value for best parameter {'criterion': 'gini', 'max_features': 'log2', 'splitter': 'best'}: 0.9331095830246935
```

3. Random Forest:

Accuracy = 0.99

ROC_AUC = 0.99

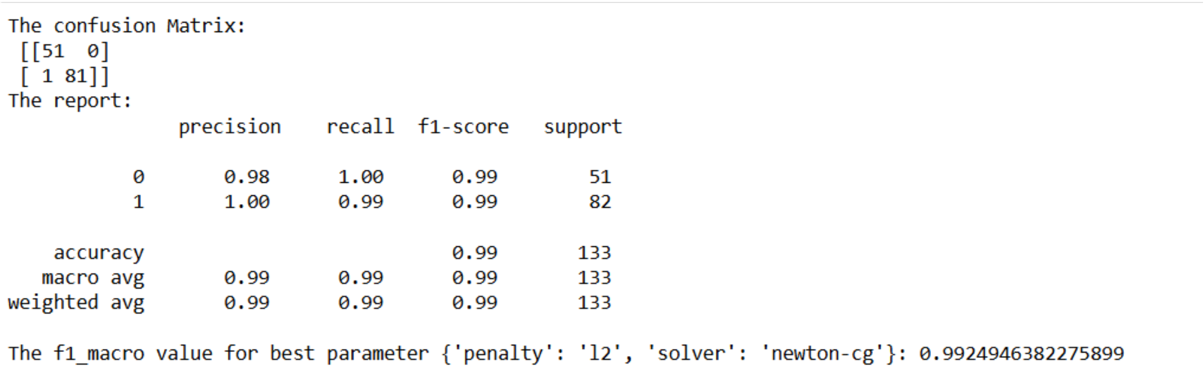


```
The confusion Matrix:
 [[51  0]
 [ 1 81]]
The report:
               precision    recall  f1-score   support

           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 f1 macro value for best parameter {'criterion': 'gini', 'max features': 'log2', 'n estimators': 100}: 0.9924946382275899
```

4. Logistic Regression:

Accuracy = 0.99

ROC_AUC = 1.0

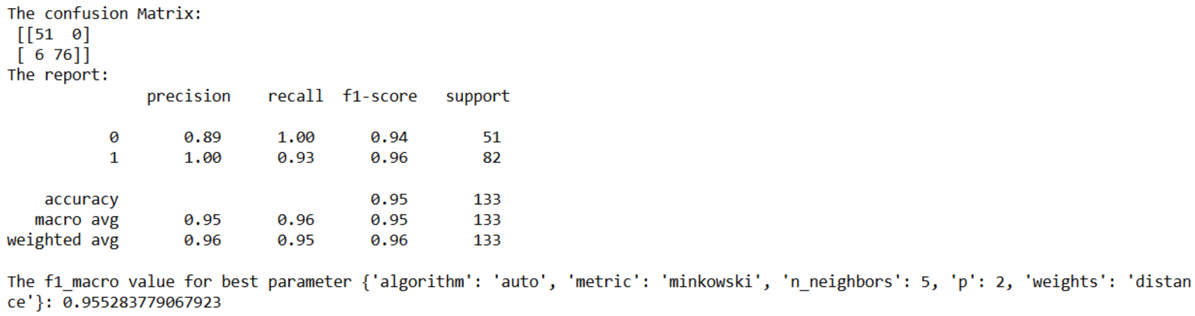


```
The confusion Matrix:
 [[51  0]
 [ 1 81]]
The report:
               precision    recall  f1-score   support

           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 f1_macro value for best parameter {'penalty': 'l2', 'solver': 'newton-cg'}: 0.9924946382275899
```

5. KNN:

Accuracy = 0.95

ROC_AUC = 1.0

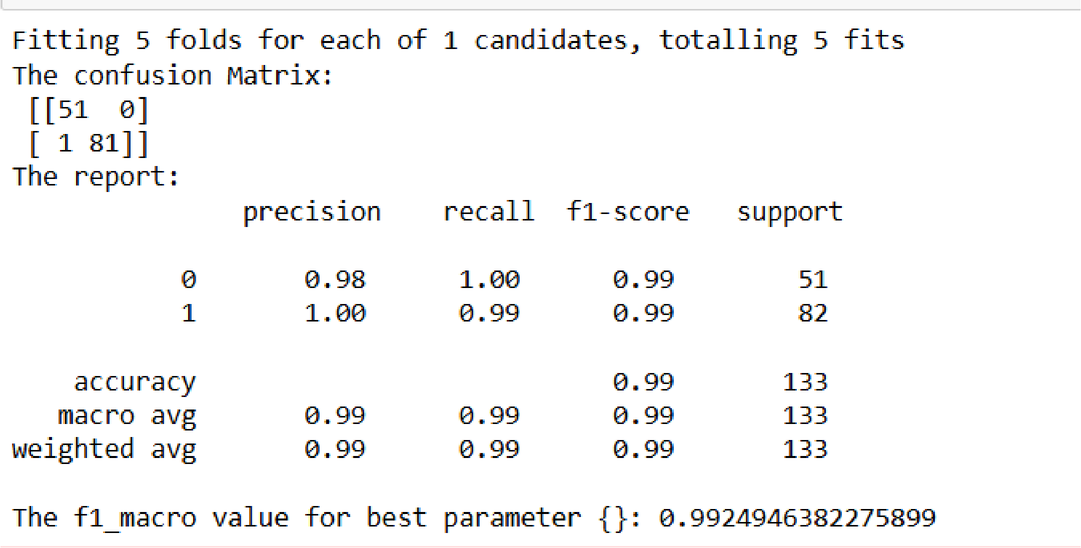


```
The confusion Matrix:
 [[51  0]
 [ 6 76]]
The report:
               precision    recall  f1-score   support

           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

The f1_macro value for best parameter {'algorithm': 'auto', 'metric': 'minkowski', 'n_neighbors': 5, 'p': 2, 'weights': 'distan
ce'}: 0.955283779067923
```

6. Navie Bayes:

Accuracy = 0.99

ROC_AUC = 0.99



```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
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 f1_macro value for best parameter {}: 0.9924946382275899
```

Final Model:

Considering the Accuracy and ROC_AUC_Score value **Logistic Regression algorithm** is the best model from the above algorithms with high accuracy and threshold value.

Accuracy = 0.99

ROC_AUC = 1.0