Hope Artificial Intelligence

Assignment-Classification

Project - Predict the Chronic Kidney Disease

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.

CKD.csv

1.Identify the Problem Statement

Stage 1: Domain selection

The input is number so it is "**Machine Learning**"

Stage 2: Learning selection

The requirement is clear and both input & output are available so it is "**Supervised**" learning

Stage 3: Supervised (Classification or Regression)

The output has categorical data so it is "**Classification**"

2.Dataset Basic Information (Total number of rows, columns)

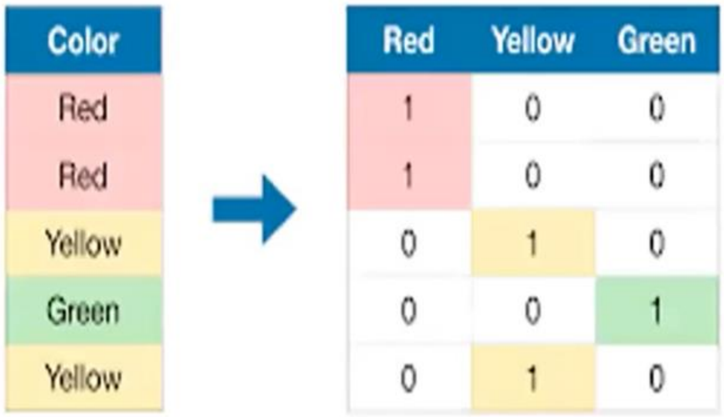
Let us see little information about client given dataset.

- Rows Detail - Total number of rows is 399
- Columns Detail – Total number of columns is 25
- We need to predict the Chronic Kidney Disease so classification column is the output
- Rest columns are our Input Parameter

3.Pre-processing method for dataset

- Here observed that rbc, pc, pcc, ba, htn, dm, cad, appet, pe, ane column looks the categorical data
- Categorical Data isn’t handle/process in AI using Python so we need to convert into meaning number
- We can define Categorical Column into Nominal and Ordinal data.
- Nominal Data
    - Data can’t compare because it isn’t in order wise.
    - For Example
        - Place Name(Erode, Chennai, Bangalore)
        - Things Name(Toys, Computer, Clothes, Door)

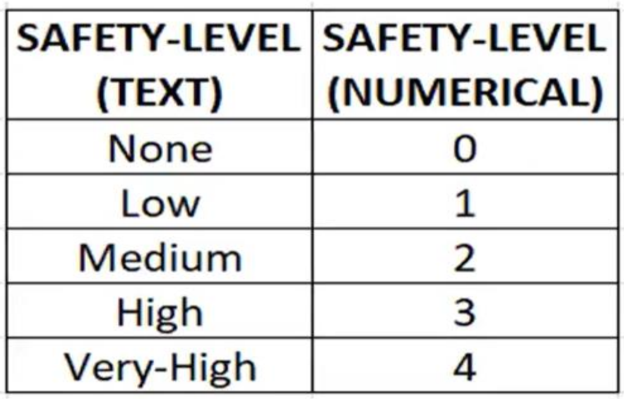
        - Use One hot encoding to convert into meaning number
        - Single Column will be expand based on column value once converted into numerical number like below



Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

- ✓ Ordinal Data
    - Ordinal Data can compare and it is in order wise
    - For Example
        - Dress Size(S,M,L,XL,XXL,XXXL)
        - Priority(High,Medium,Low)
    - Use Mapping/Label Encoder to convert into meaning number
    - Column will not expand once converted into numerical number. See below table pic.

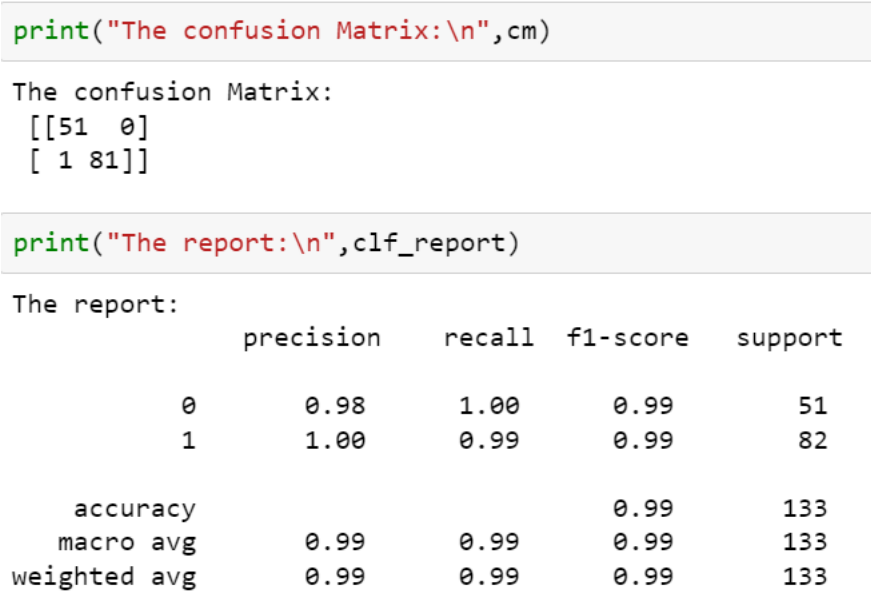


SAFETY-LEVEL (TEXT)	SAFETY-LEVEL (NUMERICAL)
None	0
Low	1
Medium	2
High	3
Very-High	4

- ✓ Here I’m go with Nominal Data conversion – String to Numerical

4.Develop a good model

1. Logistic Algorithm



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

```
The confusion Matrix:
 [[51  0]
 [ 1 81]]
```

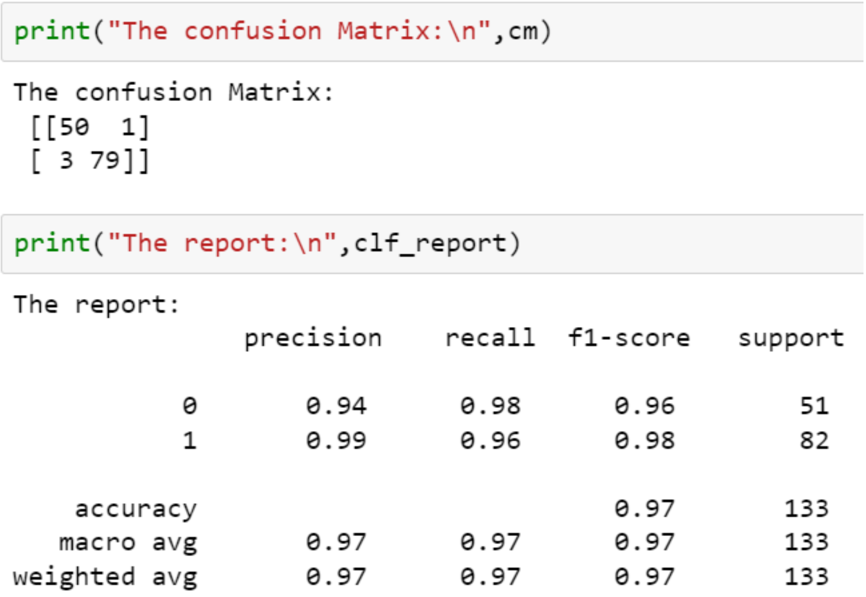
```
print("The report:\n",clf_report)
```

```
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
```

2. Decision Tree



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

```
The confusion Matrix:
 [[50  1]
 [ 3 79]]
```

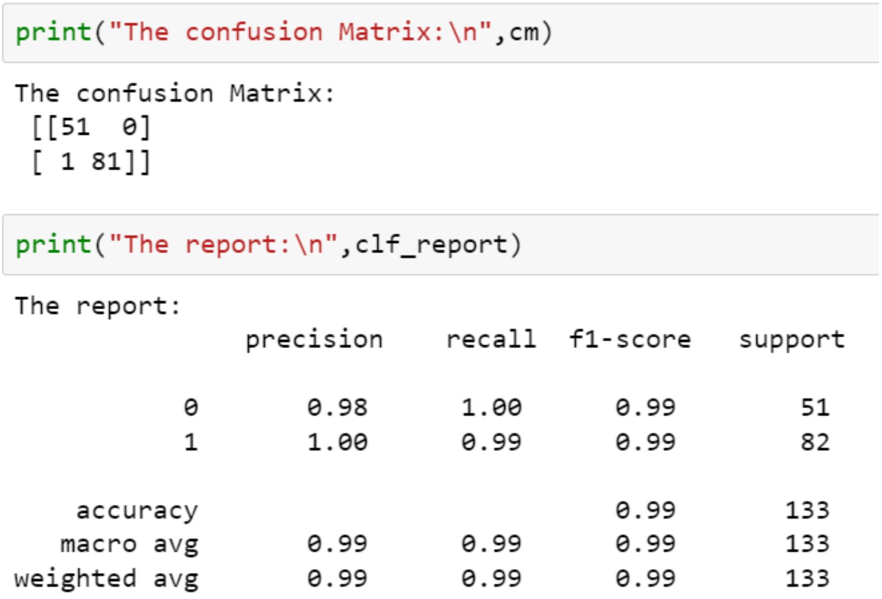
```
print("The report:\n",clf_report)
```

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

           0       0.94      0.98      0.96        51
           1       0.99      0.96      0.98        82

    accuracy                           0.97       133
   macro avg       0.97      0.97      0.97       133
weighted avg       0.97      0.97      0.97       133
```

3. Random Forest



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

```
The confusion Matrix:
 [[51  0]
 [ 1 81]]
```

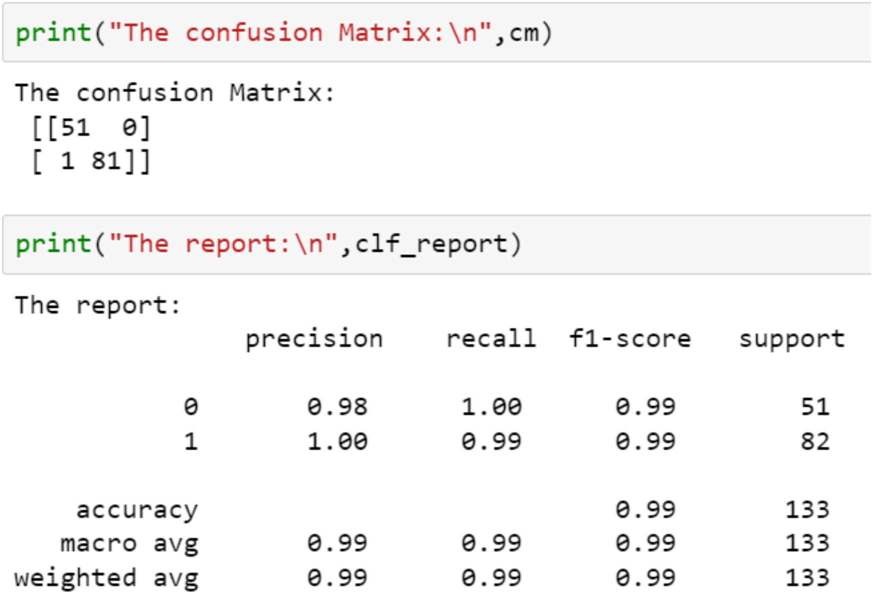
```
print("The report:\n",clf_report)
```

```
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
```

4. Support Vector Machine



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

```
The confusion Matrix:
 [[51  0]
 [ 1 81]]
```

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

```
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
```

5. **The Final Machine Learning best method of classification:**
    - Logistic Algorithm, Random Forest and Support Vector Machine give Accuracy as 0.99
6. **Justification:**
    - Why chosen the Logistic Algorithm, Random Forest and Support Vector Machine because these only given the Accuracy, Recall, Precision value as 0.99 which is too good.