CHRONIC KIDNEY DISEASE PREDICTION

1. Problem Statement

Machine Learning → Supervised Learning → Classification

2. Dataset

Number of Rows : 399

Number of Columns : 25

3. Preprocessing

To preprocess the dataset, I'm converting the categorical data into numerical format because machine learning models cannot handle it directly. Additionally, since each column has high and low values, our model would otherwise prioritize data with high values. To avoid this, I'm using a standard scaler to transform all data into a defined range. This ensures that our model gives equal importance to the data in all columns, meaning it gives equal importance to all features.

4. Models

1. Logistic Classification

```
cm
```

```
array([[32,  0],
       [ 0, 48]])
```

```
from sklearn.metrics import classification_report
clf_report=(classification_report(y_test, grid_prediction))
```

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

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

           0       1.00      1.00      1.00        32
           1       1.00      1.00      1.00        48

    accuracy                           1.00        80
   macro avg       1.00      1.00      1.00        80
weighted avg       1.00      1.00      1.00        80
```

2. RF Classification

```
Confusion Matrix:
 [[31  1]
 [ 0 48]]
```

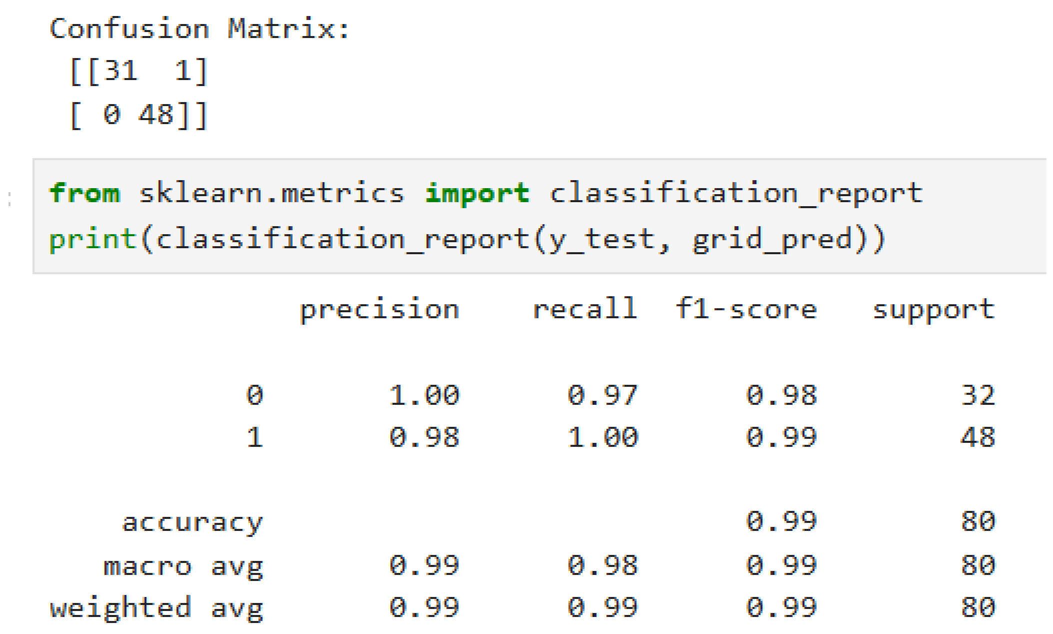
```
from sklearn.metrics import classification_report
print(classification_report(y_test, grid_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.97      0.98        32
           1       0.98      1.00      0.99        48

    accuracy                           0.99        80
   macro avg       0.99      0.98      0.99        80
weighted avg       0.99      0.99      0.99        80
```

3. Decision Tree Classification



```
Confusion Matrix:
 [[31  1]
 [ 0 48]]
```

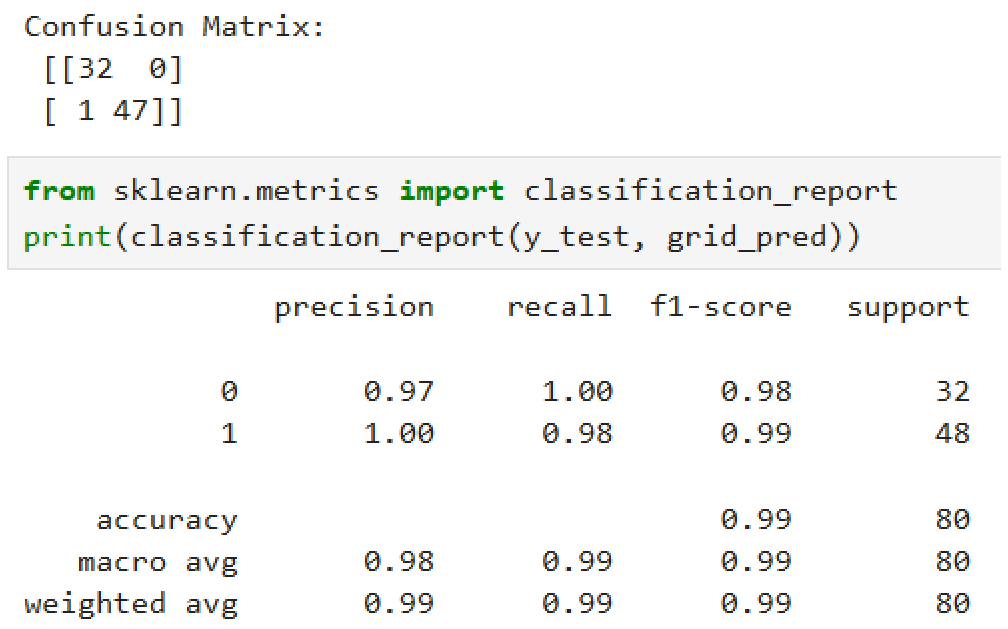
```python
from sklearn.metrics import classification_report
print(classification_report(y_test, grid_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.97      0.98        32
           1       0.98      1.00      0.99        48

    accuracy                           0.99        80
   macro avg       0.99      0.98      0.99        80
weighted avg       0.99      0.99      0.99        80
```

4. SVM Classification



```
Confusion Matrix:
 [[32  0]
 [ 1 47]]
```

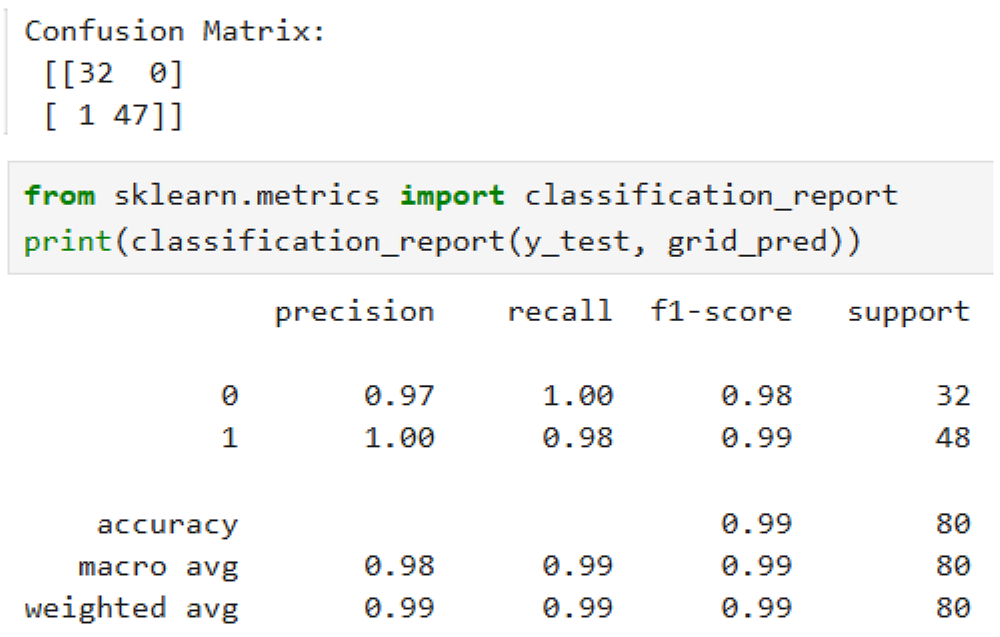
```python
from sklearn.metrics import classification_report
print(classification_report(y_test, grid_pred))
```

```
              precision    recall  f1-score   support

           0       0.97      1.00      0.98        32
           1       1.00      0.98      0.99        48

    accuracy                           0.99        80
   macro avg       0.98      0.99      0.99        80
weighted avg       0.99      0.99      0.99        80
```

5. KNN Classification



```
Confusion Matrix:
 [[32  0]
 [ 1 47]]
```

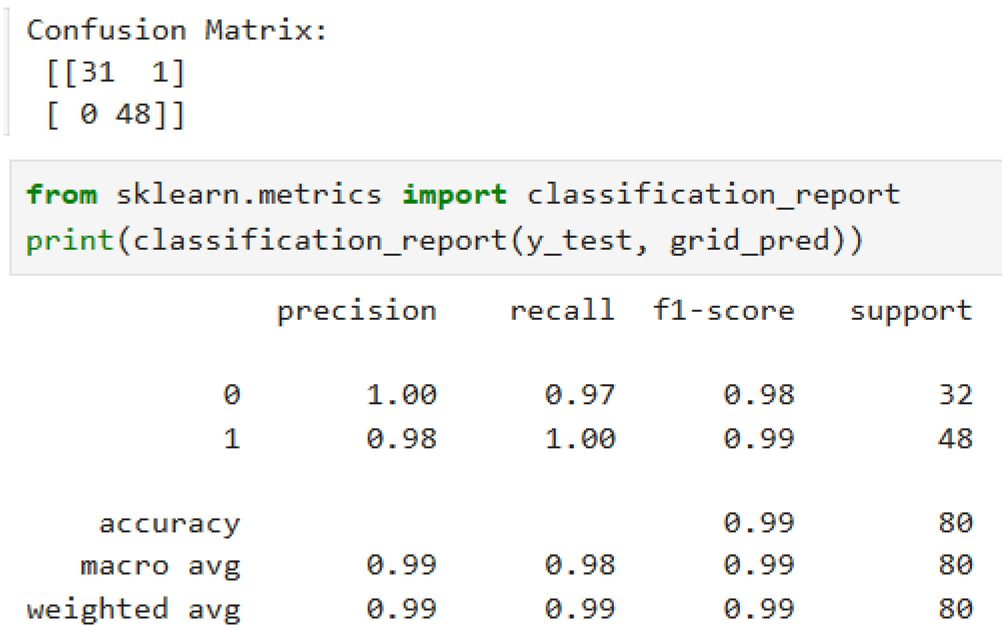
```python
from sklearn.metrics import classification_report
print(classification_report(y_test, grid_pred))
```

```
              precision    recall  f1-score   support

           0       0.97      1.00      0.98        32
           1       1.00      0.98      0.99        48

    accuracy                           0.99        80
   macro avg       0.98      0.99      0.99        80
weighted avg       0.99      0.99      0.99        80
```

6. Naïve Bayes



```
Confusion Matrix:
 [[31  1]
 [ 0 48]]
```

```python
from sklearn.metrics import classification_report
print(classification_report(y_test, grid_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.97      0.98        32
           1       0.98      1.00      0.99        48

    accuracy                           0.99        80
   macro avg       0.99      0.98      0.99        80
weighted avg       0.99      0.99      0.99        80
```

- We are developing this model to predict who has the disease. For disease prediction, recall should be high. The recall of the class we are predicting, which is the class of people who are patients, needs to be high. Why should recall be high? Because recall tells us how well our model predicts the positive values, meaning how many actual patients it identifies. It's crucial for us to correctly identify all existing positive cases. Therefore, recall is very important in this scenario. If we look at the results of all algorithms based on recall, Logistic Regression is the best. It provides the best overall output across all parameters. That's why we are selecting Logistic Regression as our best model.

  Logistic Classification : Recall = 1