Assignment – Classification with GridSearchCV Algorithm

Problem Statement or Requirement

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

Step-1: Domain Selection - Machine Learning

Step-2: Learning Method - Supervised

Step-3: Type - Classification - Categorical values as output

Step-4: convert Categorical values into numerical by nominal data method

Step-5: Try SVM, Decision Tree, Random Forest, Logistic Regression, K-Means, Navie's Bayes

Step-6: Derive the highest scoring value through confusion matrix

Step-7: Save the best model and finalize

2.) Tell basic info about the dataset (Total number of rows, columns)

Total number of Rows: 399

Total number of Columns: 25

12 categorical columns to be converted into numerical values using nominal data method

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

Yes. There are 12 categorical columns to be converted into numerical values using nominal data method.

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

1. **Support Vector Machine:** **f1-score-True = 0.99, accuracy = 0.99**
2. **Decision Tree:** **f1-score-True = 0.97, accuracy = 0.97**
3. **Random Forest:** **f1-score-True = 0.99, accuracy = 0.98**
4. **Logistic Regression:** **f1-score-True = 0.99, accuracy = 0.99**
5. **K Means:** **f1-score-True = 0.95, accuracy = 0.94**

6. Navie's Bayes:

1. Gaussian NB: f1-score-True = 0.98, accuracy = 0.98

2. Multinomial NB: f1-score-True = 0.83, accuracy = 0.82

3. Bernoulli NB: f1-score-True = 0.95, accuracy = 0.94

4. Complement NB: f1-score-True = 0.83, accuracy = 0.82

5.) All the research values (confusion_matrix of the models) should be documented. (You can make tabulation or screenshot of the results.)

1. Support Vector Machine: f1-score-True = 0.99, accuracy = 0.99

```
array([[51,  0],
       [ 1, 81]], dtype=int64)
```

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

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

       False       0.98      1.00      0.99        51
        True       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: f1-score-True = 0.97, accuracy = 0.97

```
array([[51,  0],
       [ 4, 78]], dtype=int64)
```

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

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

       False       0.93      1.00      0.96        51
        True       1.00      0.95      0.97        82

    accuracy                           0.97       133
   macro avg       0.96      0.98      0.97       133
weighted avg       0.97      0.97      0.97       133
```

3. Random Forest: f1-score-True = 0.99, accuracy = 0.98

```
array([[50,  1],
       [ 1, 81]], dtype=int64)
```

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

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

       False       0.98      0.98      0.98        51
        True       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
```

4. Logistic Regression: f1-score-True = 0.99, accuracy = 0.99

```
array([[51,  0],
       [ 1, 81]], dtype=int64)
```

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

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

       False       0.98      1.00      0.99        51
        True       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. K Means: f1-score-True = 0.95, accuracy = 0.94

```
array([[51,  0],
       [ 8, 74]], dtype=int64)
```

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

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

       False       0.86      1.00      0.93        51
        True       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
```

6. Navie's Bayes:

1. Gaussian NB: f1-score-True = 0.98, accuracy = 0.98

```
[[51  0]
 [ 3 79]]
              precision    recall  f1-score   support

       False       0.94      1.00      0.97        51
        True       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
```

2. Multinomial NB: f1-score-True = 0.83, accuracy = 0.82

```
[[50  1]
 [23 59]]
              precision    recall  f1-score   support

       False       0.68      0.98      0.81        51
        True       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
```

3. Bernoulli NB: f1-score-True = 0.95, accuracy = 0.94

```
[[51  0]
 [ 8 74]]
              precision    recall  f1-score   support

       False       0.86      1.00      0.93        51
        True       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
```

4. Complement NB: f1-score-True = 0.83, accuracy = 0.82

```
[[50  1]
 [23 59]]
              precision    recall  f1-score   support

       False       0.68      0.98      0.81        51
        True       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
```

6.) Mention your final model, justify why u have chosen the same.

Both Support Vector Machine and Logistic Regression Confusion Matrix Values of f1-score-True = 0.99, accuracy = 0.99 [Highest Score] provides the highest score compared to all model's evaluation. So, Support Vector Machine and Logistic Regression classification models can be finalized and deployed.