Machine Learning Classification – Assignment

By – J. Vinoth Sankar

Predicting Chronic Kidney Disease (CKD) based on various parameters falls under the category of Machine Learning Classification - Supervised Learning. Here's why:

Supervised Learning: This branch of machine learning deals with data that has labeled outcomes. In your case, you have a dataset with individuals as data points and the presence or absence of CKD as the labeled outcome (usually binary: 0 for no CKD, 1 for CKD). The model learns from this labeled data to make predictions about unlabeled data points, in this case, identifying individuals at risk of developing CKD.

Classification: This subcategory of supervised learning deals with predicting categorical outcomes. Since CKD is a binary classification (yes/no), it perfectly fits this category. Other examples of classification problems include spam detection, image recognition, and sentiment analysis.

Unsupervised Learning: This approach deals with unlabeled data, where the goal is to find hidden patterns or structures within the data. It wouldn't be suitable for your CKD prediction scenario because you have clear labels indicating the presence or absence of the disease. Unsupervised learning is often used for tasks like clustering documents, finding anomalies in data, or dimensionality reduction.

There are 399 rows X 25 columns in this dataset

SVM Grid Classification:

```
Best parameters: {'C': 10, 'gamma': 'auto', 'kernel': 'sigmoid'}
F1-macro score: 0.9924946382275899
Confusion matrix:
 [[51  0]
 [ 1 81]]
Classification 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

ROC AUC score: 0.9939024390243902
```

Decision Tree Grid Classification:
Best Parameters: {'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 10}

```
The f1_macro value for the best parameters is: 0.925618241407715
The confusion matrix:
 [[50  1]
 [ 9 73]]
The classification report:
              precision    recall  f1-score   support

       False       0.85      0.98      0.91        51
        True       0.99      0.89      0.94        82

    accuracy                           0.92       133
   macro avg       0.92      0.94      0.92       133
weighted avg       0.93      0.92      0.93       133

The ROC AUC score is: 0.9353180296508847
```

Here's a breakdown of the preprocessing methods used in the code:

1. Handling Categorical Features:
    - The code employs one-hot encoding for categorical features. This involves creating new binary columns for each unique value in a categorical column, indicating whether or not the original value is present.
    - The line `dataset = pd.get_dummies(dataset, drop_first=True)` accomplishes this.
2. Scaling Numerical Features:
    - Standardization is applied to numerical features to ensure they have a mean of 0 and a standard deviation of 1. This prevents features with larger scales from dominating those with smaller scales.
    - The lines `sc = StandardScaler()` and `X_train = sc.fit_transform(X_train)`, `X_test = sc.transform(X_test)` perform this scaling.
3. Handling Missing Values:
    - The code doesn't explicitly handle missing values. If your dataset contains missing values, you'll need to incorporate techniques like:
        - Imputation: Filling missing values with appropriate estimates (e.g., mean, median, mode).
        - Deletion: Removing rows or columns with missing values.
4. Handling String Data:
    - The code doesn't explicitly convert strings to numbers. If your dataset contains string features that need to be numerical for modeling, consider:
        - Label encoding: Assigning a unique integer to each distinct string value.
        - One-hot encoding: As mentioned earlier, creating binary columns for each unique string value.

Remember to tailor these preprocessing steps to the specific characteristics of your dataset and the requirements of your analysis.

The initial results using SVM Grid Classification on the CKD data are encouraging, with an accuracy of 99%. Therefore, I consider SVM Grid Classification as a final model.