SUMMARY 4

1. **Data Preprocessing and Splitting (split_scalar Function)**:
    - Imports necessary libraries and functions.
    - Splits the dataset into training and testing sets (80% training, 20% testing).
    - Standardizes the features using `StandardScaler`.
2. **PCA Dimensionality Reduction (pca Function)**:
    - Applies PCA with a specified number of components (`n`) to reduce the dimensionality of the feature space.
    - Returns the PCA model and the transformed training data.
3. **Classification with Machine Learning Algorithms**:
    - Implements classification using several machine learning algorithms, including Logistic Regression, Support Vector Machine (SVM) with linear and radial basis function (RBF) kernels, Naive Bayes, K-Nearest Neighbors (KNN), Decision Trees (DT), and Random Forest (RF).
    - For each algorithm:
        - Fits the classifier to the training data.
        - Calculates the confusion matrix and accuracy on the test data.
        - Appends the accuracy to the respective list (`acclog`, `accsvmlin`, `accsvmnl`, `accnav`, `accknn`, `accdec`, `accrand`).
4. **Results Presentation (pca_merge Function)**:
    - Combines the accuracy scores from different algorithms for each PCA component number into a DataFrame (`result`).
    - The DataFrame `result` shows the accuracy of different algorithms for different PCA component numbers.
5. **Data Loading and Preparation**:
    - Reads the "Premotorbike.csv" dataset into a Pandas DataFrame.
    - Applies one-hot encoding to categorical variables using `pd.get_dummies`.
    - Defines the feature matrix `X` and the target variable `y`.
6. **PCA Model and Dimensionality Reduction (pca_model, X_train_pca)**:
    - Applies PCA with 4 components to the feature matrix `X` and stores the PCA model and transformed data in `pca_model` and `X_train_pca`, respectively.
7. **Classification Using Various Algorithms**:

SUMMARY 4

- Applies each classification algorithm to the PCA-transformed data.
- Stores the accuracy of each algorithm in separate lists (`acclog`, `accsvmlin`, `accsvmnl`, `accnav`, `accknn`, `accdec`, `accrand`).

8. **Results Presentation (result DataFrame)**:
    - Combines the accuracy scores from different algorithms for PCA with 4 components into a DataFrame named `result`.

In summary, this code performs dimensionality reduction using PCA with 4 components and evaluates the performance of various machine learning algorithms (Logistic Regression, SVM, Naive Bayes, KNN, Decision Trees, Random Forest) on the reduced dataset. The results are presented in a DataFrame showing the accuracy of each algorithm for the reduced feature space.