INFS7203 Project Phase II README

Coding Environment

- Operation system: Window10
- Programming language: Python 3.10
- Additional packages installed: numpy, pandas, matplotlib, sklearn

Instructions

Run the main() function, the reported test and evaluation results would be reproduced. I directly use the best classifiers that have selected the hyperparameters to make predictions, data preprocessing and parameter selection functions are commented out because they need to be run step by step.

Justifications

- load_data(filename):

  The function to load data, return the feature dataframe and label dataframe.
- all_value_imputation(dataframe):

  The function to perform all value imputation and return the dataframe after imputation.
- class_specific_imputation(dataframe):

  The function to perform class-specific imputation and return the dataframe after imputation.
- knn_imputation(dataframe):

  The function to perform knn imputation and return the dataframe after imputation.
- cross_validation(X, y, model):

The function to perform 10-fold cross validation, return the mean accuracy and f1 score.

- model_based_outlier_detection(dataframe):

  The function implementing model-based outlier detection , replace outliers with the np.nan value and return the dataframe.

- density_based_outlier_detection(dataframe):

  The function implementing density-based outlier detection , replace outliers with the np.nan value and return the dataframe.

- isolation_based_outlier_detection(dataframe):

  The function implementing isolation-based outlier detection , replace outliers with the np.nan value and return the dataframe.

- max_min_normalization(dataframe):

  The function to perform max-min normalization and return the dataframe after normalization.

- standardization(dataframe):

  The function to perform standardization and return the dataframe after normalization.

- generate_conbination_csv(train_data):

  The function to generate the combination of 3 imputation methods and 3 outlier detection methods and write these results into different files.

- preprocessing_benchmark(X, y, file_name):

  The function using a decision tree classifier to get the dataset's accuracy and f1 score.

- choose_best_preprocessing_combination(file_names):

  The function to choose the best combination of preprocessing techniques.

- find_best_model(filename):

  The function to find the best classification model on the specific dataset.