CS5402 Introduction to Data Mining (SP 2025)

HW-1 (Deadline:02-20-2025, midnight)

PTB-XL, a large publicly available electrocardiography dataset

https://physionet.org/content/ptb-xl/1.0.3/

Consists of 21837 records from 18885 patients of 10 seconds length, with 12 channels. The ECG-waveform data was annotated by up to two cardiologists as a multi-label dataset, where diagnostic labels were further aggregated into super and subclasses.

Step-1: data transformation and integration.

- Load scp_codes and filename_lr from [ptbxl_sample.csv], then read [scp_statements.csv] to map scp_codes to their corresponding diagnostic_class, which will serve as labels. Retrieve the ECG signals using filename_lr, and integrate all data.
- Convert the textual class labels into one-hot encoding.
  - For example, using the label order [NORM, MI, STTC, CD, HYP], an ECG signal with labels [HYP, MI, STTC] would be converted to [0, 1, 1, 0, 1].
- and return two numpy arrays:
  - data_x with shape [num_samples, signal_length (1000), num_channels(12)]
  - data_y with shape [num_samples, num_classes (5)]

Step 2: Perform data preprocessing:

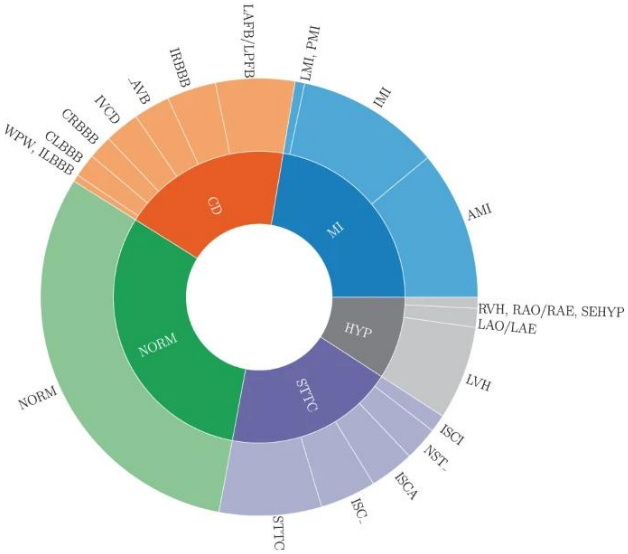
- Check for missing values (or N/A), anomalies, and outliers.
  - Fill missing values with the average of adjacent points in the same channel.
  - Replace outliers (extra-large values) using the 97th percentile (np.percentile(x, 97)).
  - Replace outliers (extra-small values) using the 3rd percentile (np.percentile(x, 3)).
- Normalize each channel with the equation: (x - xmin)/(xmax - xmin).
  - - xmax: represents the maximum value of a channel
  - - xmin: represents the minimum value of the channel.
- After normalization, the values will be scaled to range from 0 to 1.

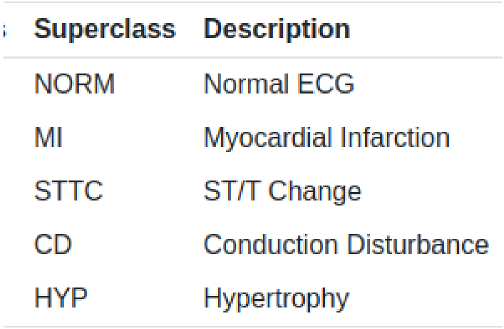
Step 3: Data split

Split the dataset into training (70%), validation (20%), and test (10%) sets.

Requirement:

1. You need to implement the following functions and strictly follow the predefined input and output formats:
   a. parse_ptbxl_data()->pd.DataFrame
   b. create_dataset(df: pd.DataFrame) -> tuple[np.ndarray, np.ndarray]
   c. data_preprocessing(data_x: np.ndarray, data_y: np.ndarray) -> tuple[np.ndarray, np.ndarray]
   d. split_data(data_x: np.ndarray, data_y: np.ndarray) -> tuple[Dict[*str*, np.ndarray], Dict[*str*, np.ndarray], Dict[*str*, np.ndarray]]
2. Only use the already imported Python libraries; any additional libraries are prohibited.
3. Ther autograder.py file can be used to test your implemented functions. Please ensure that your final submission passes the autograder tests.
4. You may use AI tools to help you understand concepts, but you must write the code yourself. Using AI-generated or AI-modified code is strictly prohibited.
5. Start early, and feel free to ask questions if you encounter any issues.





Superclass	Description
NORM	Normal ECG
MI	Myocardial Infarction
STTC	ST/T Change
CD	Conduction Disturbance
HYP	Hypertrophy

Graphical summary of the PTB-XL dataset in terms of diagnostic superclasses and subclasses.