Project structure

Data: Should contain all data files in a flat structure (no subfolders).
Continuous: Contains relevant code for the continuous wake word detection model.
Windowing: Contains relevant code for the windowing wake word detection model.

organize_files.py: Organizes the data in the data folder into three subsets, which in turn are moved in to separate subfolders.
concat_files.py: Creates new files by concatenating test files, which can be used as input to the model.

Continuous

models: Contains the parameters of trained models.
pickles: Contains preprocessed data and other data needed across separate scripts.
plots: Contains plots generated by applying the model to longer/concatenated speech files.

preprocessing.py: Preprocesses the data and stores it in separate pickle files.
model.py: Defines the TensorFlow model.
train_model.py: Trains the model previously defined and saves the parameters.
evaluate_model.py: Computes performance measures using the trained model and preprocessed test data.
plot_preds.py: Generates plots of the predicted wake word probabilities based on longer/concatenated speech files.
realtime.py: Records speech and detects wakewords in real time. After a pre-defined number of seconds, a probability plot is produced.

Windowing

models: Contains the parameters of trained models.
pickles: Contains preprocessed data and other data needed across separate scripts.
plots: Contains plots generated by applying the model to longer/concatenated speech files.

preprocessing.py: Preprocesses the data and stores it in separate pickle files.
model.py: Defines the TensorFlow model.
train_model.py: Trains the model previously defined and saves the parameters.
evaluate_model.py: Computes performance measures using the trained model and preprocessed test data.
plot_preds.py: Generates plots of the predicted wake word probabilities based on longer/concatenated speech files.