1. Sleep\_stage\_classifier
   1. All codes run within Sleep\_stage\_classifier directory
   2. Step 1: run Prepare\_model\_data.py
      1. Specify raw data location shhs\_base\_dir. It should have 2 subdirectories: edfs and annotations-events-nsrr. Within each subfolder, there is a folder shhs1 which stores the edf file and annotation file respectively
      2. Specify the output location
      3. The outputs are train/valid/test X and Y dataset saved in numpy npy format
   3. Step 2: run run\_model.py
      1. Need specify the model output folder
      2. Also need specify the PATH\_TRAIN\_FILE, PATH\_VALID\_FILE and PATH\_TEST\_FILE, which should be the same the output location as specified in step 1
      3. The output is a saved random forest model in .SAV format
      4. It will also print the F1score and accuracy of the test model results. A confusion matrix is also saved
   4. Step 3: score\_newpatient.py
      1. Need specify the location of edffile
      2. The location of X\_train: this is to standardize the scored file
      3. The location of model
      4. The location of the output folder for scored results
      5. The final output is a csv file which has sleep stage for one single patient
2. CNN\_CNN\_Model

This is an alternative model built using CNN to extract features and feed into the next CNN model. The model spec is borrowed from (<https://towardsdatascience.com/sleep-stage-classification-from-single-channel-eeg-using-convolutional-neural-networks-5c710d92d38e>) but the data preparation pipeline used our own and uses PSD instead of raw EEG data

* 1. All codes run within CNN\_CNN\_Model
  2. Step 1: run Data\_preparation.py
     1. Specify raw data location shhs\_base\_dir. It should have 2 subdirectories: edfs and annotations-events-nsrr. Within each subfolder, there is a folder shhs1 which stores the edf file and annotation file respectively
     2. Specify the output location
     3. The % of data used for model training can also be modified
     4. The output is saved CNN model in h5 format
  3. Step 2: run Train\_Model.py
     1. Specify the input location of saved npz files
     2. Specify the file name for the model and # of epochs
     3. The output is saved CNN model in h5 format in the current directory, print F1score and accuracy in the screen and save the confusion matrix
  4. Step 3: Test\_Trained\_Model.py
     1. This is a standalone program to just run the trained model on test set. The Train.Model.py runs considerable amount of time
     2. Need specify the input location of saved npz files, the saved model file name
     3. The output is a saved confusion matrix graph and f1score and accuracy printed on the screen
  5. Step 3: Scoring\_newpatient
     1. Need specify the location of edffile
     2. The name of the model
     3. The location of the output folder for scored results
     4. The final output is a csv file which has sleep stage for one single patient