Merging Json files for Karma Lego, dividing into entities and cheat for running KarmaLego project on pycharm:

After the previous manual you should have a folder with separated Json files, one for each entity. This manual presents several functions for merging Json files and expanding your training data horizontally by dividing one entity into several (for example 5-minute record into 5 records with 1 minute length).

The functions:

```
def merge_json_files(input_folder, output_file)
```

The function only merges the separate Json files to use for multi KarmaLego.

Make sure to write the full path to the both input folder and output file (add .Json in the end of output file)

```
def filter_json_sec_folders(input_folder, output_folder, sec_beginning, sec_middle, sec_end):
```

The function receives an input folder with all of the Json files and creates a new folder with new json files with only 'sec' number of seconds from beginning middle and end of each entity record. In use for saving computing power.

```
def divide_entities_and_merge(input_folder, output_file, seconds, overlap):
```

The function is made for expanding the number of entities in train and test sets by dividing to seconds with overlap.
(seconds = 30 , overlap =5 -> entitiy1part1: 0-30s ,entity1part2: 5-35s).
The function merges the files for use in multi KarmaLego

```
def divide_entities_without_merge(input_folder, output_folder, seconds, overlap):
```

Same but without merging to use in single KarmaLego.

```
def train_test_split_by_subject(df, label, subject, test_size=0.3, random_state=42):
```

After dividing each entity to parts there is danger of data leakage in regular train test split so this function create a train test split that keeps all parts of each subject in either train or test set but not divided between them.
label – the name of label column.
subject - the name of subject column.

```
def create_single_kl_configs(entities_folder, mvs, original_patterns_paths):
```

A function to create multiple single KarmaLego configs files at once and saves them in a folder she creates (where the multi KarmaLego patterns file is located) along with a folder for the future single KarmaLego results.

```
def merge_csv_files(file1_path, file2_path, output_path):
```

This function is useful for merging the data from horizontal support and mean duration .csv files.

```
def merge_x_and_y(X_file_path, y_file_path, output_folder, train_or_test,
hs_md_hsmd,label):
```

I used this function to merge my labels data with the KarmaLego features .csv. It is not so general but you can look into it and adjust it to your needs.

Cheat for running KarmaLego visual studio project on pycharm:

Locate the following variables on your compute and run this code:

```
import subprocess

# Define the path to the Visual Studio executable
visual_studio_path = r'C:\Program Files\Microsoft Visual
Studio\2022\Community\MSBuild\Current\Bin\MSBuild.exe'

# Define the path to the project file
project_path = r'C:\final_project\KarmaLego-
master\KarmaLego.Console\KarmaLego.Console.csproj'

# Define the path to the configuration files
config_file_path =
r'C:\final_project\micro_state_model\all_jsons\configs\multi_KL_config_30-
10.json'

# Define the command to run the project with the current configuration file
command = f'dotnet run --project {project_path} --config
{config_file_path}'

# Run the command in a subprocess
subprocess.run(command, shell=True)
```

You can use this code to loop and run several KarmaLego processes in a row.

For example with the config files you created with “create_single_kl_configs” function

Good luck!