Data Preprocessing:

Task: Transform JSON to pickle.

Input: dict that contains design choices as keys.

|  |  |
| --- | --- |
| Key | Description |
| ROI | The Region of Interest. Could be min/max lat/long for a rectangular region, or more complex defined region. |
| TimePeriod | The time period of interest. |
| MaxSpeed | The maximum speed we wish to include. This is done to remove erroneous and not physically possible AIS updates |
| NavStatuses | The Navigational Statuses we wish to include. Read more here.  https://help.marinetraffic.com/hc/en-us/articles/203990998-What-is-the-significance-of-the-AIS-Navigational-Status-Values- |
| Shiptypes | The AIS shiptypes we wish to include. Read more here:  https://help.marinetraffic.com/hc/en-us/articles/205579997-What-is-the-significance-of-the-AIS-Shiptype-number- |
| BinEdges | The edges of the 4-hot encoded bins. Is currently calculated in the config-file based on ROI and resolution. |
| MinTrackLength | The minimum time a track needs to exists before we include it. |
| MaxTrackLength | The maximum time a track can exists. If the track exceeds this time it is split into pieces each obeying MinTrackLength < Length < MaxTrackLength |
| ResampleFrequency | How often do we wish to resample our tracks. |

Output: Dict or better yet list of dicts that can be dumped as pickle

Step 1: Determine relevant MMIS’s to consider (Marie does not do this step)

Allocate empty dataframe “MMSIS” with columns “MMSI” and “File”

For each JSON file:

Read the “.path” data into a dataframe with columns:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Time | Lat | Long | SOG | COG |

For each timestamp in Time lookup the most recent “.statushist” and add that to a new column.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time | Lat | Long | SOG | COG | Navigation status |

Filter the dataframe based on ROI, TimePeriod, MaxSpeed, navigation status.

If rows left in dataframe and shiptype isin Shiptypes then

Add new row with current MMSI and filename to dataframe “MMSIS”

Return dataframe “MMSIS”

Step 2: Process each unique MMSI

Find all unique MMSIS in dataframe.

listOfMMSIs = pd.unique(mmsis['MMSI'])

For each mmsis in listOfMMSIs:

Step 2.1: Read and Join all files for this mmsi

Allocate list A

For each file

Convert .path and .status to dataframe similar to before

Append dataframe to list A

Concat dataframes in list A

Return dataframe and shiptype

Step 2.2: Filter dataframe

Basic filter similar to step 1

Step 2.3: Filter stationary Navigational statuses

Filter out “nonmoving” status updates

Step 2.4: Split into Tracks

If there is gap of 15 min or longer we split the updates into 2 separate trajectories

Step 2.5: Remove short Tracks

Remove tracks which are not long enough in time or have too few updates

For each track:

Step 2.6 Resample and interpolate

Step 2.7 Split long tracks into smaller segments obeying MinTrackLength < Length < MaxTrackLength

Save segment to pickle.