Preprocessing: revert shuffling, train test split, restrict and save df as csv

* Import data as strings
* Create vectors and 3d matrices of datatype np.array(vector) or list(vector, matrix).
* Invert permutate shuffle order: undoes shuffling by providing an array of indices in the order before shuffling (train, test have been shuffled simultaneously)
* Transform matrices to 3d arrays (days, sensors, timestamps)
* For each sample (day):
  + Create a dataframe with columns ‘traj\_i’ – sensor name and daily matrix transposed as data. Daily matrix: array dim(features, steps) = (rows, columns)
  + Aggregate 10 min data to hourly data (mean sensor value, groupby)
* Concatenate all days to one data frame (continuous time series over 15 month for each feature column)
* For each feature:
  + Create string key as index: feature(sensor)-sample(day)-step(hour\_of\_day) for sorting
  + Sort by index, drop index
  + Fill NaN TS\_Values: df.fillna(method=’ffill’): 1,NaN,2 -> 1,1,2
  + Create 2 additional timeseries of previous and next timestamps: df.shift(1;-1)
* Append all features to data frame
* Restrict dataset (combined train + test) to 173 days, academic reference length
* Following columns are (created and) handled as categorical: sensor id, hours from start, day of week, time of day. The reason behind is that these values can be flags for certain behavior of time series.
* Create csv