3 main routines

1. PREPARE DATA: merge, clean data frames write out
   1. Read configurations
   2. Read inputs
   3. Prepare data
      1. join the data?, calculate max irr, remove outliers?)
   4. Write output
2. TARGET DERIVATION: semi-automised derivation of anomalous behaviour
   1. Read configurations
   2. Read input
   3. Call method that calculates (e.g. method calculate\_alarms of class DataProcessor)
      1. **F1**: Cloudiness
      2. **F2**: AC to DC conversion anomalies
      3. **F3**: DC power generation alarms
         1. Calculate outliers-**F4**(input:dataframe1, variable name; output:dataframe1+new\_column) Outlier calculation: adds a column showing the points that are outlying in the inter-inverter comparison plot
         2. Calculate inverter daily efficiency-**F5**(input:dataframe1, variable name; output: new dataframe2 with inverter efficiency per inverter per day rows=inverters\*days)
         3. Derive intra-inverter alarms – **F6**(input:dataframe2, variable name; output:dataframe2+new\_columns) From the output of the function above derive the inverters that have much lower efficiency. Create a column with values=1 wherever this is true in the dataframe.

Create another column where you denote how many consecutive days has the inverter had low efficiency

* + - 1. Derive alarms for inverter sudden behaviour changes in time – **F7** (input:dataframe2, variable name; output:dataframe2+new\_columns)
      2. Derive alarms for inverter behaviour trend in time – **F8** (input:dataframe2, variable name; output:dataframe2+new\_columns)
    1. The output of this function gives two dataframes: one has the original number of rows ( inverters\* 1 row every 15 minutes); the other has onw row per inverter per day (inverters\*days)
  1. Write output dataframes in files

1. TRAINING
   1. Read configurations
   2. Read input
   3. Call train model
   4. Write output
2. PREDICTION
   1. Read configurations
   2. Read input
   3. Call predict model
   4. Write output
3. PERFORMANCE CALCULATION
   1. Read semi-outomised process output
   2. Read model prediction output
   3. Compare the predictions and create a report
   4. Write the report