Building Energy Performance Report

Generated by Doxygen 1.8.13

Contents

1 Namespace Index **1**

1.1 Packages 1

2 Hierarchical Index **3**

2.1 Class Hierarchy 3

3 Class Index **5**

3.1 Class List 5

4 File Index **7**

4.1 File List 7

5 Namespace Documentation **9**

5.1 building_energy Namespace Reference 9

5.1.1 Detailed Description 9

5.2 building_energy-master Namespace Reference 10

5.3 building_energy-master.src Namespace Reference 10

5.4 building_energy-master.src.get_data Namespace Reference 10

5.4.1 Function Documentation 10

5.4.1.1 cache_point() 10

5.4.1.2 get_point() 10

5.4.1.3 main() 11

5.5 building_energy-master.src.mv_model Namespace Reference 11

5.5.1 Function Documentation 11

5.5.1.1 create_models() 11

5.5.1.2 format_eval() . . . 12
5.5.1.3 main() . . . 12
5.5.1.4 preprocess() . . . 12
5.5.2 Variable Documentation . . . 12
5.5.2.1 info_log . . . 12
5.5.2.2 stderr . . . 13
5.5.2.3 tmy_name . . . 13
5.6 building_energy-master.src.pi_datalink Namespace Reference . . . 13
5.7 building_energy-master.src.preprocessor Namespace Reference . . . 13
5.8 building_energy-master.src.profile Namespace Reference . . . 13
5.8.1 Function Documentation . . . 13
5.8.1.1 import_test() . . . 14
5.8.2 Variable Documentation . . . 14
5.8.2.1 argv . . . 14
5.8.2.2 profiler . . . 14
5.8.2.3 stdout . . . 14
5.8.2.4 stream . . . 14
5.9 mv_model Namespace Reference . . . 14
5.9.1 Detailed Description . . . 14

6 Class Documentation **15**

6.1 building_energy-master.src.mv_model._InfoFilter Class Reference . . . 15
6.1.1 Detailed Description . . . 15
6.1.2 Constructor & Destructor Documentation . . . 15
6.1.2.1 __init__() . . . 15
6.1.3 Member Function Documentation . . . 15
6.1.3.1 filter() . . . 16
6.2 building_energy-master.src.mv_model._StreamWriter Class Reference . . . 16
6.2.1 Detailed Description . . . 16
6.2.2 Constructor & Destructor Documentation . . . 16
6.2.2.1 __init__() . . . 16

6.2.3 Member Function Documentation . . . 16
6.2.3.1 write() . . . 17
6.2.4 Member Data Documentation . . . 17
6.2.4.1 logger . . . 17
6.3 building_energy-master.src.preprocessor.DataPreprocessor Class Reference . . . 17
6.3.1 Detailed Description . . . 18
6.3.2 Constructor & Destructor Documentation . . . 18
6.3.2.1 __init__() . . . 18
6.3.3 Member Function Documentation . . . 18
6.3.3.1 add_degree_days() . . . 18
6.3.3.2 add_time_features() . . . 18
6.3.3.3 clean_data() . . . 19
6.3.3.4 count_constants() . . . 19
6.3.3.5 count_na() . . . 19
6.3.3.6 create_dummies() . . . 19
6.3.3.7 flag_data() . . . 20
6.3.3.8 flag_out_of_bound() . . . 20
6.3.3.9 flag_outliers() . . . 20
6.3.3.10 interpolate_data() . . . 20
6.3.3.11 remove_na() . . . 20
6.3.3.12 remove_out_of_bound() . . . 21
6.3.3.13 remove_outliers() . . . 21
6.3.3.14 resample_data() . . . 21
6.3.4 Member Data Documentation . . . 21
6.3.4.1 data_cleaned . . . 21
6.3.4.2 data_preprocessed . . . 21
6.3.4.3 data_raw . . . 22
6.3.4.4 data_removed . . . 22
6.3.4.5 droppedNA . . . 22
6.3.4.6 droppedOutliers . . . 22

6.3.4.7 droppedOutOfBound . . . 22
6.4 building_energy-master.src.mv_model.DataSet Class Reference . . . 22
6.4.1 Detailed Description . . . 23
6.4.2 Constructor & Destructor Documentation . . . 23
6.4.2.1 __init__() . . . 23
6.4.3 Member Function Documentation . . . 24
6.4.3.1 get_dataset() . . . 24
6.4.3.2 set_dataset() . . . 24
6.4.4 Member Data Documentation . . . 24
6.4.4.1 baseline1 . . . 24
6.4.4.2 baseline2 . . . 24
6.4.4.3 eval . . . 24
6.4.4.4 fulldata . . . 25
6.5 building_energy-master.src.mv_model.Model Class Reference . . . 25
6.5.1 Detailed Description . . . 26
6.5.2 Constructor & Destructor Documentation . . . 26
6.5.2.1 __init__() . . . 26
6.5.3 Member Function Documentation . . . 26
6.5.3.1 calc_scores() . . . 26
6.5.3.2 output() . . . 26
6.5.3.3 predict() . . . 27
6.5.3.4 project() . . . 27
6.5.3.5 train() . . . 27
6.5.4 Member Data Documentation . . . 27
6.5.4.1 baseline . . . 27
6.5.4.2 clf . . . 28
6.5.4.3 data_set . . . 28
6.5.4.4 eval . . . 28
6.5.4.5 savings . . . 28
6.5.4.6 scores . . . 28
6.6 building_energy-master.src.pi_datalink.PIDatalink Class Reference . . . 28
6.6.1 Constructor & Destructor Documentation . . . 29
6.6.1.1 __init__() . . . 29
6.6.2 Member Function Documentation . . . 29
6.6.2.1 get_stream() . . . 29
6.6.2.2 get_stream_by_point() . . . 30
6.6.2.3 get_webID_by_point() . . . 30
6.6.2.4 search_by_point() . . . 31
6.6.3 Member Data Documentation . . . 31
6.6.3.1 root . . . 31

7 File Documentation **33**

7.1 __init__.py File Reference 33

7.2 get_data.py File Reference 33

7.3 mv_model.py File Reference 33

7.4 pi_datalink.py File Reference 34

7.5 preprocessor.py File Reference 34

7.6 profile.py File Reference 35

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

building_energy
    -master::src::get_data 9
building_energy-master 10
building_energy-master.src 10
building_energy-master.src.get_data 10
building_energy-master.src.mv_model 11
building_energy-master.src.pi_datalink 13
building_energy-master.src.preprocessor 13
building_energy-master.src.profile 13
mv_model
    Main function documentation Analyzes data from sensors in UC Davis buildings 14

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

building_energy-master.src.mv_model._StreamWriter 16
Filter
    building_energy-master.src.mv_model._InfoFilter 15
object
    building_energy-master.src.mv_model.DataSet 22
    building_energy-master.src.mv_model.Model 25
    building_energy-master.src.pi_datalink.PIDatalink 28
    building_energy-master.src.preprocessor.DataPreprocessor 17

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

building_energy-master.src.mv_model._InfoFilter 15
building_energy-master.src.mv_model._StreamWriter 16
building_energy-master.src.preprocessor.DataPreprocessor 17
building_energy-master.src.mv_model.DataSet 22
building_energy-master.src.mv_model.Model 25
building_energy-master.src.pi_datalink.PIDatalink 28

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

__init__.py 33
get_data.py 33
mv_model.py 33
pi_datalink.py 34
preprocessor.py 34
profile.py 35

Chapter 5

Namespace Documentation

5.1 building_energy Namespace Reference

-master::src::get_data

5.1.1 Detailed Description

-master::src::get_data

-master::src::profile

-master::src::preprocessor

-master::src::pi_datalink

```
This file tests getting Data from the PI_datalink stream


This class is a wrapper for PI web API

It replicates the functions of PI datalink excel Add-on (Windows only) in Python

Some features may be UC Davis specific, but can easily be extended to other PI installations

@author Marco Pritoni <marco.pritoni@gmail.com>
latest update: Feb 14 2017

TODO:
-return a df from summary
--from API
-pass dic from search to make retrieve data quicker for multi-points
-use batch API calls for several datapoints at the same time
-add path search


Preprocesses and cleans data.

@author Marco Pritoni <marco.pritoni@gmail.com>
latest update: Feb 14 2017

TODO:

-think about best way of interpolating/resampling (1-groupby TimeGrouper, resample, interpolate)
-save the list of point removed
-add preprocessing as in mave (data normalization around 0)

version 0.1


This file checks the runtime of the program
```

5.2 building_energy-master Namespace Reference

Namespaces

- src

5.3 building_energy-master.src Namespace Reference

Namespaces

- get_data
- mv_model
- pi_datalink
- preprocessor
- profile

5.4 building_energy-master.src.get_data Namespace Reference

Functions

- def cache_point (point_name, start="2014", end="t", remove_duplicates=True)
- def get_point (point_names, start="2014", end="t")
- def main ()

5.4.1 Function Documentation

5.4.1.1 cache_point()

```
def building_energy-master.src.get_data.cache_point (
              point_name,
              start = "2014",
              end = "t",
              remove_duplicates = True )
```

```
Gets the stream and saves the data in CSV format

Parameters:
point_name: name of data point
start: start parameter
end: end parameter
remove_duplicates: flag
```

5.4.1.2 get_point()

```
def building_energy-master.src.get_data.get_point (
              point_names,
              start = "2014",
              end = "t" )
```

```
Gets the data point from the data stream

Parameters:
point_names: Names of data points
start: start parameter
end: end parameter
```

5.4.1.3 main()

```
def building_energy-master.src.get_data.main ( )
```

```
Main function. Initializes sample data point to cache
```

5.5 building_energy-master.src.mv_model Namespace Reference

Classes

- class _InfoFilter
- class _StreamWriter
- class DataSet
- class Model

Functions

- def main ()
- def create_models (args=None)
- def preprocess (data)
- def format_eval (data, tmy_data, tmy_slice, input_vars, output_vars)

Variables

- stderr
- info_log = open("../logs/info.log", "w")
- string tmy_name = "NSRDB.136708.OAT.TMY"

5.5.1 Function Documentation

5.5.1.1 create_models()

```
def building_energy-master.src.mv_model.create_models (
              args = None )
```

Create Measurement Verification models and prints data. Default args uses sys.argv

5.5.1.2 format_eval()

```
def building_energy-master.src.mv_model.format_eval (
              data,
              tmy_data,
              tmy_slice,
              input_vars,
              output_vars )
```

Formats the evaluation data using the TMY data and time period.

```
Parameters:
data: Full data
tmy_data: Data from the TMY period
tmy_slice: TMY time period
input_vars: Input variables
output_vars: Output variables
```

5.5.1.3 main()

```
def building_energy-master.src.mv_model.main ( )
```

Main function. Starts the logger and creates the models

5.5.1.4 preprocess()

```
def building_energy-master.src.mv_model.preprocess (
              data )
```

Cleans and sets up the data to be used

```
Parameters:
data: Data values
```

5.5.2 Variable Documentation

5.5.2.1 info_log

```
building_energy-master.src.mv_model.info_log = open("../logs/info.log", "w")
```

5.5.2.2 stderr

```
building_energy-master.src.mv_model.stderr
```

5.5.2.3 tmy_name

```
string building_energy-master.src.mv_model.tmy_name = "NSRDB.136708.OAT.TMY"
```

5.6 building_energy-master.src.pi_datalink Namespace Reference

Classes

- class PIDatalink

5.7 building_energy-master.src.preprocessor Namespace Reference

Classes

- class DataPreprocessor

5.8 building_energy-master.src.profile Namespace Reference

Functions

- def import_test ()

Variables

- profiler = line_profiler.LineProfiler(import_test)

  Profile object instance.
- stdout
- argv

  Command line arguments.
- stream

  Data stream.

5.8.1 Function Documentation

5.8.1.1 import_test()

```
def building_energy-master.src.profile.import_test ( )
```

```
Import function for profiling. Starts temporary logging and imports appropriate libraries
```

5.8.2 Variable Documentation

5.8.2.1 argv

```
building_energy-master.src.profile.argv
```

Command line arguments.

5.8.2.2 profiler

```
building_energy-master.src.profile.profiler = line_profiler.LineProfiler(import_test)
```

Profile object instance.

5.8.2.3 stdout

```
building_energy-master.src.profile.stdout
```

5.8.2.4 stream

```
building_energy-master.src.profile.stream
```

Data stream.

5.9 mv_model Namespace Reference

Main function documentation Analyzes data from sensors in UC Davis buildings.

5.9.1 Detailed Description

Main function documentation Analyzes data from sensors in UC Davis buildings.

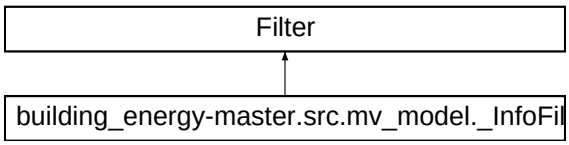
Trains, projects, and calculates scores and savings.

Chapter 6

Class Documentation

6.1 building_energy-master.src.mv_model._InfoFilter Class Reference

Inheritance diagram for building_energy-master.src.mv_model._InfoFilter:



Public Member Functions

- def __init__ (self)
- def filter (self, record)

6.1.1 Detailed Description

```
Filter to allow only INFO level messages to appear in info.log
```

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def building_energy-master.src.mv_model._InfoFilter.__init__ (
              self )
```

6.1.3 Member Function Documentation

6.1.3.1 filter()

```
def building_energy-master.src.mv_model._InfoFilter.filter (
              self,
              record )
```

Filters the messages

The documentation for this class was generated from the following file:

- mv_model.py

6.2 building_energy-master.src.mv_model._StreamWriter Class Reference

Public Member Functions

- def __init__ (self, name=__name__)
- def write (self, message)

Public Attributes

- logger

6.2.1 Detailed Description

Custom logger to wrap around file streams

6.2.2 Constructor & Destructor Documentation

6.2.2.1 __init__()

```
def building_energy-master.src.mv_model._StreamWriter.__init__ (
              self,
              name = __name__ )
```

6.2.3 Member Function Documentation

6.2.3.1 write()

```
def building_energy-master.src.mv_model._StreamWriter.write (
              self,
              message )
```

6.2.4 Member Data Documentation

6.2.4.1 logger

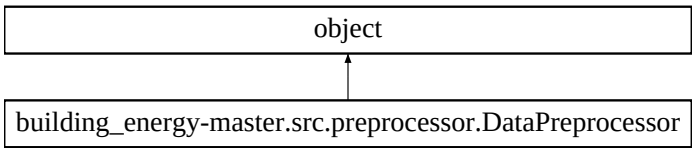
```
building_energy-master.src.mv_model._StreamWriter.logger
```

The documentation for this class was generated from the following file:

- mv_model.py

6.3 building_energy-master.src.preprocessor.DataPreprocessor Class Reference

Inheritance diagram for building_energy-master.src.preprocessor.DataPreprocessor:



Public Member Functions

- def __init__ (self, df, args, kwargs)
- def resample_data (self, data, freq)
- def interpolate_data (self, data, interpolate_limit)
- def remove_na (self, data, na_how)
- def remove_outliers (self, data, sd_val)
- def remove_out_of_bound (self, data, low_bound, high_bound)
- def clean_data (self, resampling=True, freq="h", interpolating=True, interpolate_limit=1, removing_↩ na=True, na_how="any", removing_outliers=True, sd_val=3, enforcing_bounds=True, low_bound=0, high_↩ bound=9998, args, kwargs)
- def flag_data (self, runRemoveNA=True, removeNAhow="any", runRemoveOutliers=True, sd_val=3, run↩ RemoveOutOfBound=True, low_bound=0, high_bound=9998, runExtendIndex=False)
- def flag_outliers (self, data, sd_val)
- def flag_out_of_bound (self, data, low_bound, high_bound)
- def count_na (self, data)
- def count_constants (self, data)
- def add_time_features (self, data)
- def add_degree_days (self, data, hdh_cpoint=65, cdh_cpoint=65)
- def create_dummies (self, data, var_to_expand=["TOD", DOW)

Public Attributes

- data_raw
- data_cleaned
- data_removed
- data_preprocessed
- droppedNA
- droppedOutliers
- droppedOutOfBound

6.3.1 Detailed Description

```
Preprocessor class for data cleaning and manipulation
(standardization for machine learning)
```

6.3.2 Constructor & Destructor Documentation

6.3.2.1 __init__()

```
def building_energy-master.src.preprocessor.DataPreprocessor.__init__ (
              self,
              df,
              args,
              kwargs )
```

6.3.3 Member Function Documentation

6.3.3.1 add_degree_days()

```
def building_energy-master.src.preprocessor.DataPreprocessor.add_degree_days (
              self,
              data,
              hdh_cpoint = 65,
              cdh_cpoint = 65 )
```

6.3.3.2 add_time_features()

```
def building_energy-master.src.preprocessor.DataPreprocessor.add_time_features (
              self,
              data )
```

6.3.3.3 clean_data()

```
def building_energy-master.src.preprocessor.DataPreprocessor.clean_data (
              self,
              resampling = True,
              freq = "h",
              interpolating = True,
              interpolate_limit = 1,
              removing_na = True,
              na_how = "any",
              removing_outliers = True,
              sd_val = 3,
              enforcing_bounds = True,
              low_bound = 0,
              high_bound = 9998,
              args,
              kwargs )
```

6.3.3.4 count_constants()

```
def building_energy-master.src.preprocessor.DataPreprocessor.count_constants (
              self,
              data )
```

counts the % of points in each TS that does not change

6.3.3.5 count_na()

```
def building_energy-master.src.preprocessor.DataPreprocessor.count_na (
              self,
              data )
```

6.3.3.6 create_dummies()

```
def building_energy-master.src.preprocessor.DataPreprocessor.create_dummies (
              self,
              data,
              var_to_expand = ["TOD",
              DOW )
```

6.3.3.7 flag_data()

```
def building_energy-master.src.preprocessor.DataPreprocessor.flag_data (
              self,
              runRemoveNA = True,
              removeNAhow = "any",
              runRemoveOutliers = True,
              sd_val = 3,
              runRemoveOutOfBound = True,
              low_bound = 0,
              high_bound = 9998,
              runExtendIndex = False )
```

6.3.3.8 flag_out_of_bound()

```
def building_energy-master.src.preprocessor.DataPreprocessor.flag_out_of_bound (
              self,
              data,
              low_bound,
              high_bound )
```

6.3.3.9 flag_outliers()

```
def building_energy-master.src.preprocessor.DataPreprocessor.flag_outliers (
              self,
              data,
              sd_val )
```

6.3.3.10 interpolate_data()

```
def building_energy-master.src.preprocessor.DataPreprocessor.interpolate_data (
              self,
              data,
              interpolate_limit )
```

6.3.3.11 remove_na()

```
def building_energy-master.src.preprocessor.DataPreprocessor.remove_na (
              self,
              data,
              na_how )
```

6.3.3.12 remove_out_of_bound()

```
def building_energy-master.src.preprocessor.DataPreprocessor.remove_out_of_bound (
              self,
              data,
              low_bound,
              high_bound )
```

6.3.3.13 remove_outliers()

```
def building_energy-master.src.preprocessor.DataPreprocessor.remove_outliers (
              self,
              data,
              sd_val )
```

```
Removes all data data above or below sd_val standard deviations
from the mean and excludes all lines with NA in any column
```

6.3.3.14 resample_data()

```
def building_energy-master.src.preprocessor.DataPreprocessor.resample_data (
              self,
              data,
              freq )
```

6.3.4 Member Data Documentation

6.3.4.1 data_cleaned

```
building_energy-master.src.preprocessor.DataPreprocessor.data_cleaned
```

6.3.4.2 data_preprocessed

```
building_energy-master.src.preprocessor.DataPreprocessor.data_preprocessed
```

6.3.4.3 data_raw

building_energy-master.src.preprocessor.DataPreprocessor.data_raw

6.3.4.4 data_removed

building_energy-master.src.preprocessor.DataPreprocessor.data_removed

6.3.4.5 droppedNA

building_energy-master.src.preprocessor.DataPreprocessor.droppedNA

6.3.4.6 droppedOutliers

building_energy-master.src.preprocessor.DataPreprocessor.droppedOutliers

6.3.4.7 droppedOutOfBound

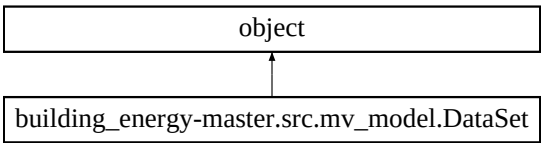
building_energy-master.src.preprocessor.DataPreprocessor.droppedOutOfBound

The documentation for this class was generated from the following file:

- preprocessor.py

6.4 building_energy-master.src.mv_model.DataSet Class Reference

Inheritance diagram for building_energy-master.src.mv_model.DataSet:



Public Member Functions

- def __init__ (self, data, tPeriod1=(slice(None)), tPeriod2=(slice(None)), tPeriod3=(slice(None)), out=[""], inp=[""])
- def set_dataset (self, baseline_type, date_slicer, inpt, outpt)
- def get_dataset (self, baseline_type, date_slicer, inpt_outpt)

Public Attributes

- fulldata

  Stores all the data.
- baseline1

  Stores the data associated with baseline 1 period.
- baseline2

  Stores the data associated with baseline 2 period.
- eval

  Stores the data associated with evaluation period.

6.4.1 Detailed Description

```
Inspired by Paul Raftery Class:
fist prototype

the dataset_type field is to help standardize notation of different datasets:
       "A":"measured pre-retrofit data",
       "B":"pre-retrofit prediction with pre-retrofit model",
       "C":"pre-retrofit prediction with post-retrofit model",
       "D":"measured post-retrofit data",
       "E":"post-retrofit prediction with pre-retrofit model",
       "F":"post-retrofit prediction with pos-tretrofit model",
       "G":"TMY prediction with pre-retrofit model",
       "H":"TMY prediction with post-retrofit model"
typical comparisons used by mave:
    Pre-retrofit model performance = A vs B
    Single model M&V = D vs E
    Post retrofit model performance  = D vs F
    Dual model M&V, normalized to tmy data = G vs H

Parameters:
data: Data from the database
tPeriod1: First time period
tPeriod2: Second time period
tPeriod3: Third time period
out: Stores output data
inp: Stores input data
```

6.4.2 Constructor & Destructor Documentation

6.4.2.1 __init__()

```
def building_energy-master.src.mv_model.DataSet.__init__ (
              self,
              data,
              tPeriod1 = (slice(None)),
              tPeriod2 = (slice(None)),
              tPeriod3 = (slice(None)),
              out = [""],
              inp = [""] )
```

6.4.3 Member Function Documentation

6.4.3.1 get_dataset()

```
def building_energy-master.src.mv_model.DataSet.get_dataset (
              self,
              baseline_type,
              date_slicer,
              inpt_outpt )
```

6.4.3.2 set_dataset()

```
def building_energy-master.src.mv_model.DataSet.set_dataset (
              self,
              baseline_type,
              date_slicer,
              inpt,
              outpt )
```

6.4.4 Member Data Documentation

6.4.4.1 baseline1

```
building_energy-master.src.mv_model.DataSet.baseline1
```

Stores the data associated with baseline 1 period.

6.4.4.2 baseline2

```
building_energy-master.src.mv_model.DataSet.baseline2
```

Stores the data associated with baseline 2 period.

6.4.4.3 eval

```
building_energy-master.src.mv_model.DataSet.eval
```

Stores the data associated with evaluation period.

6.4.4.4 fulldata

`building_energy-master.src.mv_model.DataSet.fulldata`

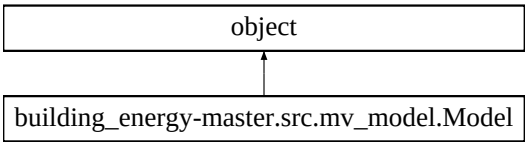
Stores all the data.

The documentation for this class was generated from the following file:

- mv_model.py

6.5 building_energy-master.src.mv_model.Model Class Reference

Inheritance diagram for building_energy-master.src.mv_model.Model:



Public Member Functions

- def __init__ (self, model_type, data_set=None)
- def train (self, baseline, out_var)
- def project (self, eval_data, out_var)
- def predict (self, data)
- def output (self)

Static Public Member Functions

- def calc_scores (compare, p, out_var)

Public Attributes

- clf
- data_set

  DataSEt object created from data.
- baseline

  Baseline period data.
- eval

  Evaluation period data.
- savings

  Calculated enery savings.
- scores

  Calculated R2 scores.

6.5.1 Detailed Description

```
Measurement Verification Model.

Parameters:
model_type: String that describes the model type
data_set: DataSet used to fit model and create projection

Attributes:
clf: Instance of the specified model type
data_set: DataSet object created from data
baseline: Baseline period data
eval: Evaluation period data
savings: Energy savings calculated
scores: Calculated R2 scores
```

6.5.2 Constructor & Destructor Documentation

6.5.2.1 __init__()

```
def building_energy-master.src.mv_model.Model.__init__ (
              self,
              model_type,
              data_set = None )
```

6.5.3 Member Function Documentation

6.5.3.1 calc_scores()

```
def building_energy-master.src.mv_model.Model.calc_scores (
              compare,
              p,
              out_var )  [static]

Calculates the R2 scores.

Parameters:
compare: A two column dataframe with one column with output variable and one with the model prediction
p: p is the number of variables in the model (eg. count the columns in the dataframe with input variables)
out_var: Output
```

6.5.3.2 output()

```
def building_energy-master.src.mv_model.Model.output (
              self )

prints model outputs and relevant statistics
```

6.5.3.3 predict()

```
def building_energy-master.src.mv_model.Model.predict (
              self,
              data )
```

Predicts on the data.

Parameters:
data: Data to predict on

6.5.3.4 project()

```
def building_energy-master.src.mv_model.Model.project (
              self,
              eval_data,
              out_var )
```

Predicts in the period specified by eval_data. The energy differences between the model and actual data is also predicted. These values are stored into savings.

Parameters:
eval_data: Data in the evaluation period
out_var: Ouput

6.5.3.5 train()

```
def building_energy-master.src.mv_model.Model.train (
              self,
              baseline,
              out_var )
```

Trains the model using baseline period data.

Parameters:
baseline: A dictionary with keys "in" and "out" that map to a pandas DataFrame

6.5.4 Member Data Documentation

6.5.4.1 baseline

```
building_energy-master.src.mv_model.Model.baseline
```

Baseline period data.

6.5.4.2 clf

`building_energy-master.src.mv_model.Model.clf`

6.5.4.3 data_set

`building_energy-master.src.mv_model.Model.data_set`

DataSEt object created from data.

6.5.4.4 eval

`building_energy-master.src.mv_model.Model.eval`

Evaluation period data.

6.5.4.5 savings

`building_energy-master.src.mv_model.Model.savings`

Calculated enery savings.

6.5.4.6 scores

`building_energy-master.src.mv_model.Model.scores`

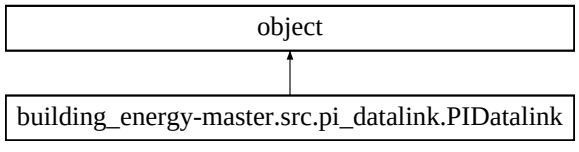
Calculated R2 scores.

The documentation for this class was generated from the following file:

- mv_model.py

6.6 building_energy-master.src.pi_datalink.PIDatalink Class Reference

Inheritance diagram for building_energy-master.src.pi_datalink.PIDatalink:



Public Member Functions

- def __init__ (self, root=None, calculation=None, interval=None, buildingMeterDB=None)
- def search_by_point (self, point_query, dataserver="s09KoOKByvc0-uxyvoTV1UfQVVRJTC1QSS1Q", include_WebID=True)
- def get_webID_by_point (self, point_name, dataserver="s09KoOKByvc0-uxyvoTV1UfQVVRJTC1QSS1Q")
- def get_stream_by_point (self, point_names, _start="y", _end="t", _calculation="interpolated", _interval="1h", _sumType=None, _label=None, dataserver="s09KoOKByvc0-uxyvoTV1UfQVVRJTC1QSS1Q", WebID_↩ dic=None)
- def get_stream (self, Web_ID=None, _start="y", _end="t", _calculation="interpolated", _interval="1h", _↩ sumType=None, _label=None)

Public Attributes

- root

6.6.1 Constructor & Destructor Documentation

6.6.1.1 __init__()

```
def building_energy-master.src.pi_datalink.PIDatalink.__init__ (
              self,
              root = None,
              calculation = None,
              interval = None,
              buildingMeterDB = None )
```

6.6.2 Member Function Documentation

6.6.2.1 get_stream()

```
def building_energy-master.src.pi_datalink.PIDatalink.get_stream (
              self,
              Web_ID = None,
              _start = "y",
              _end = "t",
              _calculation = "interpolated",
              _interval = "1h",
              _sumType = None,
              _label = None )
```

```
This method gets the stream given a WebID. It works with one stream at the time.

arguments:
Web_ID=None : - the unique identifier of the time series
_start="y" : - start date, default yesterday "y"; can use different formats as "YYYY-MM-DD";
_end="t" : - end date, default yesterday "t"; can use different formats as "YYYY-MM-DD";
_calculation="interpolated": can use "recorded" to get raw data and summary to get summary data (tot, mean, sc
note: summary data is not a time series, but a dictionary
_interval="1h": interpolation interval, used only with interpolated; default 1 hour
_sumType=None : used if calculation is "summary", can use All, Total, default Total
_label=None : not used at the moment; needed for future extensions

returns:
DataFrame object for TS
dictionary for summary
single value for end
```

6.6.2.2 get_stream_by_point()

```
def building_energy-master.src.pi_datalink.PIDatalink.get_stream_by_point (
              self,
              point_names,
              _start = "y",
              _end = "t",
              _calculation = "interpolated",
              _interval = "1h",
              _sumType = None,
              _label = None,
              dataserver = "s09KoOKByvc0-uxyvoTV1UfQVVRJTC1QSS1Q",
              WebID_dic = None )

This method gets the stream given a the point name.
It calls get_webID_by_point to get the Web ID by point name
Then it calls the stream using the Web ID
It also works with multiple points

arguments:
point_names : name or list of PI point names
_start="y" : - start date, default yesterday "y"; can use different formats as "YYYY-MM-DD";
_end="t" : - end date, default yesterday "t"; can use different formats as "YYYY-MM-DD";
_calculation="interpolated": can use "recorded" to get raw data and summary to get summary data (tot, mean, sc
note: summary data is not a time series, but a dictionary
_interval="1h": interpolation interval, used only with interpolated; default 1 hour
_sumType=None : used if calculation is "summary", can use All, Total, default Total
_label=None : not used at the moment; needed for future extensions

returns:
DataFrame object for TS
dictionary for summary
single value for end
```

6.6.2.3 get_webID_by_point()

```
def building_energy-master.src.pi_datalink.PIDatalink.get_webID_by_point (
              self,
              point_name,
              dataserver = "s09KoOKByvc0-uxyvoTV1UfQVVRJTC1QSS1Q" )

This method is to make sure we get a single WebID as result of the get_stream_by_point search
```

6.6.2.4 search_by_point()

```
def building_energy-master.src.pi_datalink.PIDatalink.search_by_point (
              self,
              point_query,
              dataserver = "s09KoOKByvc0-uxyvoTV1UfQVVRJTC1QSS1Q",
              include_WebID = True )
```

```
This method searches for points allowing * operators. It returns point list and a Dictionary with names:WebIDs

arguments:

point_query: name expression
dataserver: default point to UC Davis
include_WebID: by default True, ut returns list AND a dictionary {name : Web_ID, ...}


Example API json returned:

{
  "Items": [
    {
      "WebId": "P09KoOKByvc0-uxyvoTV1UfQ61oCAAVVRJTC1QSS1QXFBFUy5BSFUuQ09PTElORyBFTkVSR1kgQlRVIFBFUiBIUg",
      "Name": "PES.AHU.Cooling Energy BTU per Hr"
    },
    {
      "WebId": "P09KoOKByvc0-uxyvoTV1UfQ7FoCAAVVRJTC1QSS1QXFBFUy5BSFUuSEVBVElORyBFTkVSR1kgQlRVIFBFUiBIUg",
      "Name": "PES.AHU.Heating Energy BTU per Hr"
    }
}

returns:
It returns a list with point names and a dictionary with name : Web_ID
```

6.6.3 Member Data Documentation

6.6.3.1 root

```
building_energy-master.src.pi_datalink.PIDatalink.root
```

The documentation for this class was generated from the following file:

- pi_datalink.py

Chapter 7

File Documentation

7.1 __init__.py File Reference

Namespaces

- building_energy-master.src

7.2 get_data.py File Reference

Namespaces

- building_energy-master.src.get_data
- building_energy
  -master::src::get_data

Functions

- def building_energy-master.src.get_data.cache_point (point_name, start="2014", end="t", remove_↩ duplicates=True)
- def building_energy-master.src.get_data.get_point (point_names, start="2014", end="t")
- def building_energy-master.src.get_data.main ()

7.3 mv_model.py File Reference

Classes

- class building_energy-master.src.mv_model.DataSet
- class building_energy-master.src.mv_model.Model
- class building_energy-master.src.mv_model._StreamWriter
- class building_energy-master.src.mv_model._InfoFilter

Namespaces

- building_energy-master.src.mv_model
- mv_model

  Main function documentation Analyzes data from sensors in UC Davis buildings.

Functions

- def building_energy-master.src.mv_model.main ()
- def building_energy-master.src.mv_model.create_models (args=None)
- def building_energy-master.src.mv_model.preprocess (data)
- def building_energy-master.src.mv_model.format_eval (data, tmy_data, tmy_slice, input_vars, output_vars)

Variables

- building_energy-master.src.mv_model.stderr
- building_energy-master.src.mv_model.info_log = open("../logs/info.log", "w")
- string building_energy-master.src.mv_model.tmy_name = "NSRDB.136708.OAT.TMY"

7.4 pi_datalink.py File Reference

Classes

- class building_energy-master.src.pi_datalink.PIDatalink

Namespaces

- building_energy-master.src.pi_datalink
- building_energy

  -master::src::get_data

7.5 preprocessor.py File Reference

Classes

- class building_energy-master.src.preprocessor.DataPreprocessor

Namespaces

- building_energy-master.src.preprocessor
- building_energy

  -master::src::get_data

7.6 profile.py File Reference

Namespaces

- building_energy-master.src.profile
- building_energy

  -master::src::get_data

Functions

- def building_energy-master.src.profile.import_test ()

Variables

- building_energy-master.src.profile.profiler = line_profiler.LineProfiler(import_test)

  Profile object instance.
- building_energy-master.src.profile.stdout
- building_energy-master.src.profile.argv

  Command line arguments.
- building_energy-master.src.profile.stream

  Data stream.