FLASK Forecasting Server Documentation

General Information

Server File Structure:

DATA_FOLDER/station/"files"

Server Request

response = requests.get('SERVERURL/COMMAND/STATIONNAME', params=PARAMETERDICT)

where:

SERVERURL: Server adress

COMMAND: train, forecast, usw… for reference see section "commands"

STATIONNAME: name of the station

PARAMETERDICT: dict of parameters required for the command

Server Reply

The forecasting server replies with a string, which can be encoded as a json object. This json object itself contains three variables:

STATUS: either True or False, depending on the successful or not successful outcome of the request

VALUE: If available, the result of the request as a float

MESSAGE: A string containing a little more detailed information about the outcome of the request.

These values can be retrieved as following:

response.json().get("message") , or with "value" resp. "status"

Model Configuration

The model definitions are saved in a json format file and have to be uploaded as “config.json” to the station folder prior to using the forecasting function. A configuration file example is shown below. It contains the definitions for the two modeltypes “monthly” and “decadal”. Use http://jsonlint.com/ to validate the syntax of the configuration file.

```
{
  "monthly":{
    "datasets":["runoff","runoff","temp","precip"],
    "leadtimes":[[1,3],[-24,-1],[-24,-1],[-24,-1]],
    "scoremethod":"soviet_longterm",
    "fillnan":true,
                "modeltype":"RF",
                "modelparam":{
                                        "n_estimators":100
                                        }
  },
  "decadal":{
    "datasets":["runoff","runoff","temp","precip"],
    "leadtimes":[[0,0],[-24,-1],[-24,-1],[-24,-1]],
    "scoremethod":"soviet_shortterm",
    "fillnan":true,
                "modeltype":"RF",
                "modelparam":{
                                        "n_estimators":100
                                        }
  }
}
```

Configuration Options

Datasets: Contains the names of datasets used in the model. The first one in the list is by definition the target dataset. Names can be chosen freely, but must suit the filenames of the data .csv file uploaded later.

Leadtimes: Define the beginning and end of the time windows used for each dataset in the forecast relative to the forecasting date. The unit is “decades”. Positive values refer to future decades, negative values accordingly to past decades. The order is the same as in the list of Datasets. Therefore both list must have the same length. The first pair is by definition the time window over which the target value is averaged.

Scoremethod: This parameters defines how the score during crossvalidation is calculated. Options are “R2”,”soviet_longterm” and “soviet_shortterm”. The latter can only be used for one step ahead forecasts.

Fillnan: If set to true, missing values in a feature time series are linearly interpolated as long as the number of missing values is below 7.5% of the total number of values.

Modeltype: The options are: RF (random forests) and EF (extra forests)

Modelparam: Parameters according to the Skilearn documentation for the specified Modeltype.

Commands

Creating and deleting a new station

Description: Creates or deletes a folder in the database for this station. Caution: delete will erase all files uploaded to this folder.

Command: new, delete

Parameters: None

Example: requests.get('http://0.0.0.0:5000/new/pskem')

Uploading files to a station

Variant 1

Description: A file object is passed to the server through the POST request. Only .csv and .json files can be uploaded. Old files are overwritten.

Command: upload

Parameters: 'file': fileobject

Example: requests.post('http://0.0.0.0:5000/upload/pskem', files={'file': open('/home/jules/test.csv, 'rb')})

Variant 2

Description: An URL and a filename is passed through the GET request method. The server will download the file contained in the URL and save it according to the filename. The filename parameters must end with .csv or .json.

Command: upload

Parameters: 'url': url_to_file

'filename': filename

Example: requests.get('http://0.0.0.0:5000/upload/pskem', params= {'url': 'https://dl.dropboxusercontent.com/u/140367/bla.csv', 'filename':'test.csv' })

Training of the model(s)

Description: This command trains the model with all available data. The model is saved to the database folder. Older models will be overwritten. If no modeltype is specified, all modeltypes defined in the configuration file will be trained.

Command: train

Parameters: 'type': optional, modeltype according to the types defined in the config file for this station

Example: requests.get('http://0.0.0.0:5000/train/pskem', params= {'type': 'monthly' })

Crossvalidation of a model

Description: Returns the score of a 10-fold cross validation over the available dataset.

Command: crossvalidate

Parameters: 'type': modeltype according to the types defined in the config file for this station.

Example: requests.get('http://0.0.0.0:5000/crossvalidate/pskem', params= {'type': 'monthly' })

Receiving a forecast

Description: Returns the model forecast. If data have been uploaded after the last training request, these data will be used in the forecast, but the model will not be retrained. This has to be triggered manually

Command: forecast

Parameters: 'type': modeltype according to the types defined in the config file for this station.

'date': optional, string in the format YYYYMMDD e.g. 20160401. The date is automatically converted to the decadal date. When no date is given, a forecast for today will be returned. If the database already contains a value for the requested date, the server will return the database value. In that case the variable status of the reply will be set to False.

Example: requests.get('http://0.0.0.0:5000/forecast/pskem', params= {'type': 'monthly', 'date':'20160425})