Prepare the predictive features for the machine-learning models

Kai-Uwe Eiselt

12.09.2025

The scripts used in the following are to be found in the following Python libraries:

```
Python_Avalanche_Library/Python_Avalanche_PreProcessing
```

The functions and predefined lists, dictionaries and path can be found in the package `ava_functions` that must be installed using

```
pip install .
```

in the directory `Ava_Functions`.

1 NORA3

To download the NORA3 the `metocean-api` Python package was adjusted. A Github repository with the adjusted scripts exists (`NORA3_Download_Packages`) and can be made available on request. The instructions given there show how to download the northern Norwegian subset of the data used in the publications.

In the new/current methodology grid cells are extracted for several elevation bands (0-300, 300-600, 600-900, 900-1200m) and then aggregated (see below). To do this first find the grid cells in these elevation bands with (note that this was already done for in the NORA3 predictor preprocessing in is only included here for completeness):

```
python Find_Grid_Cells_Above_or_Between_CertainHeight.py --reg_code 3009
 --h_low 0 --h_hi 300
```

The execution must be repeated for the other elevation bands and regions. If, for some reason, for the further procedure one does not have the Geopandas library available (as in my case on the Climateserver) one can convert the resulting .shp files to .csv files with

```
python Convert_Height_shps_to_DataFrame.py --reg_code 3009
```

```
 --h_low 0 --h_hi 300
```

This must be repeated for all regions and elevation bands.

The next step is to in fact extract the grid cells found in the last step. This can be done with

```
Batch_Extract_NORA3_GridCells_Between_NC.py
```

In the default version this extracts the grid-cell data for all the above-defined elevation bands and for the years 1970 through 2024 individually for each year because of large memory requirements. This can be changed in the script.

The resulting data must then be concatenated with

```
Batch_Concat_Extracted_NORA3_GridCell_Data_NC.py
```

Again, this does the concatenation for all the above defined elevation bands and for the years 1970 to 2024. This can be changed in the script. This produces the netCDF files in the directory `Avalanche_Region_Data`.

A further preprocessing step is the splitting of the precipitation into rain and snow, which is here done on the hourly value basis using the temperature threshold of 0 °C:

```
Batch_Calc_rain_snow.py
```

The default script performs the calculations for all the above-mentioned elevation bands. The script updates the netCDF files in the `Avalanche_Region_Data`.

The predictors can now be calculated (also using a batch script):

```
Batch_Calc_and_Store_Predictors_NORA3.py
```

In its default state this script generates that predictors for the five warning regions and all the above-defined elevation levels for the full NORA3 time series, i.e., currently from 1970 to 2024 (directory: `Avalanche_Predictors_Full_TimeSeries`).

The predictive features can now be combined with the danger levels:

```
Batch_Gen_Preds_With_DangerLevels.py
```

and then the danger levels can be aggregated to the 2 (default) or 4 danger levels:

```
Batch_Gen_Store_XLevel_Balanced_Predictors.py
```

For each warning region then execute the following script to aggregate the predictive features across elevation bands:

```
Aggregate_NORA3_Predictors_Across_ElevBands.py
```

For the later hindcasting of the avalanche danger for the whole NORA3 period (currently 1970-2024) also aggregate the full time-series predictors with:

```
Aggregate_NORA3_Full_TimeSeries_Predictors_Across_ElevBands.py
```

This concludes the generation of the NORA3-based predictive features. Now for the SNOWPACK-derived features.

Since the SNOWPACK model either requires long-wave radiation or surface temperature, which are not available from NORA3, a procedure is provided to generate a linear model to predict the surface temperature from tas, wind speed, long- and short-wave radiation:

```
ERA5_Ts_Tas_Radiation_Model.py
```

to be found in the `Snowpack` directory. The following scripts are located in the subdirectory `NorCP`. The surface temperature can then by calculated with

```
Calc_NORA3_Ts_From_ERA5_StatModel.py
```

and the existing files can be updated with the ts data using

```
Update_NORA3_with_Ts.py
```

Note that these scripts are in

```
Python_Avalanche_Library/Snowpack/NORA3/NORA3_Ts
```

The NORA3 data can now be prepared for the SNOWPACK simulations using:

```
Prep_NORA3_Data_for_SNOWPACK.py
```

This must be repeated for every region and elevation band. Subsequently, SNOWPACK can be executed with:

```
Execute_SNOWPACK.py
```

The SNOWPACK outputs can then be further processed to find weak layers and generate time series of parameters and indices with:

```
Prep_Indices_from_SNOWPACK_Output.py
```

(There is also a batch script available for this.) Depending on the available memory, this procedure cannot be performed for the whole time series at a time and must thus be divided into slices. To subsequently concatenate the slices use:

```
Concatenate_SNOWPACK_Indices.py
```

Note that the methodology is in this case to run SNOWPACK for time slices, perform the index preparation on the indices and then to concatenate the indices.

This concludes the calculations for the NORA3 and NORA3-SNOWPACK features.

2 NorCP

The necessary variables to be downloaded from the NorCP archive are:

- tas, pr, uas, vas, hurs, rlds, rsds

Note that to be able to download the NorCP data, access must be personally requested (`https://sites.google.com/met.no/downscaling/data`). After the download these data should be regridded to the NORA3 grid, which can be done with the script `Regrid_NorCP_to_NORA3_Grid.py` (using CDO) that can be executed in a batch process with `Call_Regrid_NorCP_to_NORA3_Grid.py`.

After the regridding the snow and rain data should be calculated from the precipitation variable `pr`, which can be done with `Gen_Rain_Snow_NorCP.py`. Then, wind speed and direction should be calculated from `uas` and `vas` using `Gen_Wind_Speed_Dir_NorCP.py`.

In the new/current methodology grid cells are extracted for several elevation bands (0-300, 300-600, 600-900, 900-1200m) and then aggregated (see below). To do this first find the grid cells in these elevation bands with (note that this was already done for in the NORA3 predictor preprocessing in is only included here for completeness):

```
python Find_Grid_Cells_Above_or_Between_CertainHeight.py --reg_code 3009
 --h_low 0 --h_hi 300
```

The execution must be repeated for the other elevation bands and regions. If, for some reason, for the further procedure one does not have the Geopandas library available (as in my case on the Climateserver) one can convert the resulting .shp files to .csv files with

```
python Convert_Height_shps_to_DataFrame.py --reg_code 3009
 --h_low 0 --h_hi 300
```

This must be repeated for all regions and elevation bands.

The next step is to in fact extract the grid cells found in the last step. This can be done with

```
Extract_NorCP_GridCells_Between_NC.py --model EC-Earth --scen rcp45
 --period MC --low 0 --high 300
```

For this script a batch procedure is provided in `Batch_Extract_NorCP_GridCells.py`, as this extraction must be performed for all models, scenarios, periods and elevation bands (the default version of the script extracts for all northern Norwegian regions).

Due to memory requirements the script extracts and stores data for individual years and thus the data must be subsequently concatenated with

```
Concat_Extracted_NorCP_GridCell_Data_NC.py --model EC-Earth --scen rcp45
 --period MC --low 0 --high 300
```

(No batch script for this for now.)

This concludes the data preparation and enables the calculation of the avalanche predictors using:

```
Calc_and_Store_NorCP_Predictors.py --reg_code 3009 --agg_type mean
 --model EC-Earth --scen rcp45 --period MC --low 0 --high 300
```

From the concatenated avalanche region data, the SNOWPACK input data can be generated. The scripts for the SNOWPACK-related procedures are located in

```
Python_Avalanche_Library/Snowpack
```

The data to run the SNOWPACK model can then be prepared with

```
Prep_NorCP_Data_for_SNOWPACK.py --reg_code 3009 --low 0 --high 300
 --model EC-Earth --scen rcp45 --period MC
```

A batch procedure for this is provided in

```
Batch_Prep_NorCP_Data_for_SNOWPACK.py
```

This enables the execution of SNOWPACK, for which a procedure is provided in

```
Execute_SNOWPACK.py
```

Note that this has to be done individually for all the elevation bands and regions. The SNOWPACK output can then be post-processed to generate the stability indices with

```
Prep_Indices_from_SNOWPACK_Output.py --reg_code 3009 --low 0 --high 300
 --source NorCP --model EC-Earth --scen rcp45 --period MC
```

A batch procedure for this is provided in

```
Batch_Prep_Indices_from_SNOWPACK_Output.py
```

Both the NorCP- and SNOWPACK-derived predictive features are now available for the different elevation bands. An aggregation procedure that takes the maximum or minimum (depending on what is appropriate) across the elevation bands is provided separately for the NorCP and and SNOWPACK features in

```
Aggregate_NorCP_Predictors_Across_ElevBands.py
```

and

```
Aggregate_SNOWPACK_Stab_Inds_Across_ElevBands.py
```

For both batch procedures are available in

```
Batch_Aggregate_NorCP_Predictors_Across_ElevBands.py
```

and

```
Batch_Agg_SNOWPACK_Stab_Inds_Across_ElevBands.py
```

With this, the data to apply that machine-learning model to predict avalanche danger and avalanche days is fully available and the model can be applied, e.g., with

```
Train_Model_With_SNOWPACK.py
```

which can be found in

```
Python_Avalanche_Library/Python_Avalanche_Prediction/
```