III. Model documentation and write-up

Information included in this section may be shared publicly with challenge results. You can respond to these questions in an e-mail or as an attached file. Please number your responses.

1. Who are you (mini-bio) and what do you do professionally? If you are on a team, please complete this block for each member of the team.

   Evgenii Malygin:
   I graduated from the Department of Hydrology at Lomonosov Moscow State University in 2015. Then in 2018 I graduated from Lomonosov MSU Postgraduate School of Geosciences. For the last 2 years I have been working as a data scientist at MegaFon (top cellular operator in Russia) at the department of Big Data Analytics and Machine Learning. In my spare time, I do about the same things I do at work: I develop machine learning models and analyze data in various competitions and hackathons.

   Maksim Kharlamov:
   I am hydrologist and junior researcher in Lomonosov Moscow State University. At work, I am developing ML forecasts for extreme hydrological events: floods, droughts, ice jams, etc. My best friends are meteorological and satellite data. In love with CatBoost.

   Ivan Malygin:
   I've graduated from Mathematics faculty of Lomonosov Moscow State University, the Chair of intellectual systems. I work professionally in data analysis and modeling in various fields, such as economics and geosciences.

   Maria Sakirkina:
   Professional cartographer (Lomonosov Moscow State University, Russia), I work mainly with geographic information systems, I have been processing remote sensing data for forests, agriculture, and other natural resources for 10 years. In addition to data processing and visualization, I do analysis, the last couple of years I have been focusing more on data science methods. I teach my own course on geodata visualization and analysis at the university.

   Ekaterina Rets:
   14 years' experience in Hydrology, Glaciology and Civil Engineering. PhD in Hydrology since 2013. Currently employed at Institute Institute of Geophysics Polish Academy of Sciences. My research interests have been focused on functioning of river catchments and response of hydrological systems to climate change across different scales from small alpine catchments to mountain systems and vast regions. Re-established and led field hydrological investigation in the Djankuat Alpine Research Catchment (North Caucasus) since 2007. Developed the AMelt energy-balance model of snow and ice melt in alpine areas. Co-developer of the grwat R-package for the automatic hydrograph separation and hydrological time series analysis. Contributing author of the IPCC 6th Assessment Report (WGII, Chapter 4: Water). Has prepared as the first author publications in top international journals such as ESSD (Top 10%, IF=11.3) and Climatic Change (IF=4.7).

2. What motivated you to compete in this challenge?

   Evgenii Malygin:

   I like to participate in competitions that are close to research tasks, where you have to think, get into the problem, engineer features, and not just stacking XGBboost. It was a great challenge at the intersection of hydrology, Earth science, and machine learning.

   Maksim Kharlamov:
   I am a fan of competitions and hackathons, and this task is a perfect mix of ML, data engineering and hydrology! So, this hackathon is a great opportunity to apply maximum knowledge from various fields to solve this task. Hope I can use this experience in my scientific research.

   Ivan Malygin:
   As a hobby, I enjoy data mining and machine learning competitions. This allows me to be in trend of modern science and algorithms and use the experience gained in my main work.

   Maria Sakirkina:
   An interesting challenge related to the professional profile of the participants. The need to prepare the data from scratch, automating the process from downloading the data to obtaining the result of the prediction.

   Ekaterina Rets:
   Estimation of the spatial distribution of snow water equivalent in mountainous areas is currently one of the most important unsolved problems in snow hydrology. That makes it fascinating to contribute my knowledge to try to solve this task for the Western US. I was excited to carry out that research in a very strong interdisciplinary team (Evgeniy Malygin, Maxim Kharlamov, Ivan Malygin, Maria Sakirkina) and compete with specialists all around the Globe.

3. High level summary of your approach: what did you do and why?

   Here we present a combined physically-based and machine learning approaches to SWE prediction at 1 km resolution over the Western U.S. using different near real-time data sources (in-situ, remote sensing data, general circulation modeling data, DEM) developed in course of the Snowcast Showdown Drivendata.org competition hosted by Bureau of Reclamation.
   The solution is based on different SOTA implementations of Gradient Boosting Machine algorithm: XGBoost, LightGBM, and CatBoost, and their ensembles. The target dataset based on SNOTEL, CDEC and ASO LiDAR data was provided by the contest organizers. The top features of 121 features set included ground snow measure data (SNOTEL, CDEC), and remote sensing of snow cover (MODIS Terra MOD10A1). The top 1-4 features include

as well physically-based indirect predictors of SWE: seasonal cumulative sum of solid precipitation, seasonal average values of air temperature and the mean seasonal value of solar radiation. Terrain parameters characterizing spatial differences in incoming solar radiation display substantial level of importance, especially aspect characteristics. Integration of the energy balance snow model was tested but not included in the final solution due to low calculation speed exceeding the contest limitations (8 hours inference time). An end-to-end solution and automated real-time forecast pipeline were developed to reproduce the forecast for each week.

4. Do you have any useful charts, graphs, or visualizations from the process?

   All information, including graphs, charts, and visualizations was provided in the model report. You can download it here.

5. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.

   we parallelize even groupby for best code performance in meteorological data processing

```
def applyParallel(dfGrouped, func, dates_str):
    retLst = Parallel(n_jobs=-1)(delayed(func)(group, dates_str) for name,
group in tqdm(dfGrouped))
    return pd.concat(retLst)

features = applyParallel(df.groupby('cell_id'), cell_id_f, dates_str)
```

   We create a lot of features. In this example we provide DEM features calculation

```
def get_raster_stats(dem_buff_file, tmp_dem='processed/dem_tmp/'):
    fullpath = os.path.join(tmp_dem, dem_buff_file)
    fullpath = str(Path(fullpath))

    radius = int(dem_buff_file.split('_')[-1].split('.')[0])
    idx = dem_buff_file.split('_')[0]
    dem = rd.LoadGDAL(fullpath, no_data=-9999)
    slope = rd.TerrainAttribute(dem, attrib='slope_degrees')
    aspect = rd.TerrainAttribute(dem, attrib='aspect')
    curv_prof = rd.TerrainAttribute(dem, attrib='profile_curvature')
    curv_plan = rd.TerrainAttribute(dem, attrib='planform_curvature')
    curv = rd.TerrainAttribute(dem, attrib='curvature')
    gdal.DEMProcessing(os.path.join(tmp_dem, 'tri_' + dem_buff_file),
gdal.Open(fullpath), 'TRI', computeEdges=True)
    tri = gdal.Open(os.path.join(tmp_dem, 'tri_' +
dem_buff_file)).ReadAsArray()

    out = pd.concat([array_stats(dem, idx, 'alt', radius),
    array_stats(slope, idx, 'slope', radius),
    array_stats(aspect, idx, 'aspect', radius),
    array_stats(curv_prof, idx, 'curv_prof', radius),
```

```
    array_stats(curv_plan, idx, 'curv_plan', radius),
    array_stats(curv, idx, 'curv', radius),
    array_stats(tri, idx, 'tri', radius),
    ], axis=1)

    return out
```

we use RandomForest for interpolation of ground measures SWE data. It works much better than Kriging or IDW interpolation

```
X_org=org[['latitude','longitude', 'elevation_m', 'year', 'dayofyear']]
y_org=org['org_value']

rf=RandomForestRegressor(n_estimators=250, random_state=0, n_jobs=-1)
rf.fit(X_org,y_org)
rf_int=pd.DataFrame(rf.predict(Z_org))
```

6. Please provide the machine specs and time you used to run your model.

   We used 3 workstations for data analysis, feature engineering, model training and experiments. All calculations were performed on the CPU, the GPU was not used. Specifications are in the table below:

	Workstation 1	Workstation 2	Workstation 3
CPU	AMD Ryzen Threadripper 1950X	AMD Ryzen 9 5950X	Intel i7 11800H
CPU cores	16	16	8
CPU threads	32	32	16
RAM, gb	64	32	16
GPU	2x Nvidia GeForce 1080Ti 11gb	Nvidia GeForce RTX 2070 8 gb	Nvidia GeForce RTX 3060 6 gb
OS	Windows 10	Windows 10	Windows 10

   Train and inference time:

   - Collecting DEM features: about 1 hour on Workstation 2
   - Collecting train dataset time: about 27 hours on Workstation 3
   - Training time: about 30 min on Workstation 2
   - Inference time: about 10 min on Workstation 2 for each submission date

7. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?
   See more in README.md

8. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?

   No

9. How did you evaluate performance of the model other than the provided metric, if at all?

   No

10. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?

    We tried to use openAMUNDSEN snow and hydroclimatological modeling framework.
    The model required the creation of a separate dataset and calibration for each region (setting temperature and precipitation gradients and other physical parameters that affect the processes of SWE evolution), which complicates the calculations. openAMUNDSEN is written in Python and has a low calculation speed. For example, one section of the Sierra Nevada region for one year was calculated about 9 hours. The inference time of the model was limited to 8 hours.
    To characterize snow interception by vegetation land use and soil data (Land Cover Gridded Map, FAO-UNESCO Global Soil Regions Map) was incorporated in the first steps of model training, and further removed due to low feature importance.

11. If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?

    - Feature engineering: add SWE time-lags, try to add additional features
    - Try to use neural networks