**NOTES:**

Started working with V5. Though I have scripts in GEE to sanitize the shape files, I chose to do it in QGIS this time.

Remember that the actual version of the data is outside final. Everything has been copied to final.

Filtered by year. Josef’s ‘preprocess\_by\_shape’ hasn’t been done – for now I am not caring about multipolygons etc.

**Imp!** Buffering hasn’t been done as well. Do it in the future.

**Imp!** Getting the satellite data from getDownloadURL() - I am ignoring the metadata in the tiff files. I would like to take care of this in the future. Right now I am using txt files to store the band names (timesteps)

**UPDATE:** No longer using getdownloadurl. Simply a img.sample in a multiprocessing environment.

**Im1 -**  The proper way to run these scripts would be to change the run option in spyder to run in CWD instead of renaming CWD to the directory of the file, which is the default behaviour. Make sure you realize this, lest you get stuck on errors when executing some other project codes. (Actually, changing the seetings for the codes in this project shouldn’t change other codes – you have to manually define the run options for every file in spyder. So technically, you shouldn’t get any errors as long as you make sure every file in this project has CWD as FINAl for imports to work properly)

**IMP!** Hm doesn’t have nulls in unique\_id but optical.csv does (parallel text to csv 2 ). It shouldn’t – this looks like its coming from sjoin, where 2 lat long pairs don’t belong to any polygon. Check kthis later.

Be careful about global sowing dates, base temps etc with remapping target. I think remapping should come after, since original names might be important for base temps and everything.

Why should you even loop inputs? Create multiple instances of the run and do them parallel. Merge outputs when necessary. **NO,**  It would be useless to run all preprocessing steps just to run another ML model .

At some point MASK\_CLDPRB was changed to CLDPRB

CLDPRB itself has nan values. Check it.

**IMP** Implementing pipeline\_executables in every file. I think this imposes that all the optional parameters be in config, and that it never saves files. Save has to be done in the main file. Currently this either has to read or execute, but there should be 3rd option, to ignore.

Because of the change of YYYYMMDD\_\_ , you will get some bugs in the existing version of the code. Its okay, though.

-> Be careful about whether you are stating timesteps from 0 or 1. Especially in feature addition this is important.

The functionality for aggregating points of the same field has been removed from gdd script.

Cloud probability can exist even if all other bands are null. This is coded into the gdd function by just checking for len(bands)-1 along with len(bands)

Even in 20 day interval, GDD gets nulls if cldprb>75 is masked.

Currently having to copy everytime data is being sent into the pipeline. This creates a lot of redundancy.

Removing last columns that had null values.

Could the gdd error be because of passing dataframes as arguments?

Why does harmonised time composite have fixed variable?

**FUTURE WORK**

Things I still need to do – Cloudcheck , Estimate sow dates, filter sow date true and do ML , EDA. Experiment with Borutapy etc.

**-> bands by 1000**

**=>** Merge global vars and config in the future maybe.

**IMP** The code should be such that multiple input files are read and separate pipelines are started for all of them, including where to merge them. For instance, several years, or optical and sar. They may not be merged at all, with different ML results for them, or merged at a specific point before ML.

Somewhere it needs to be saved - the relation between timesteps to dates (for ex. In harmonised\_time\_composite)

++++++++++++++

List of things to check later on

1. That your time columns are of the pattern YYYYMMDD\_\_BAND\_NAME

I would like to strictly impose this. This isn’t being followed currently.

2. If any columns don’t exist, the data has to change, not the config file. For example, the column ‘Sow\_Date’ is used in a lot of places. So is Unique\_Id (this was field\_id before)

3. The yaml file could have errors. When your code is using the values, it is better to write an entire if else chain ( including else and raising an error, or do a check on the yaml values)

4. Use locks on data such that you can run multiple instances of your code.

5. Visualizations for Data analytics – I would like to experiment with how data is different based on provider, sow date etc. both before and after ML.

6. Saving outputs, configurations. Automate several runs with different configs. Create a config wrapper that can override certain parameters. Maybe you can create a Queue of runs that keep running in the background.

7. **IMP** Config should have all the paths. Also, path syntaxes should change, like capitalization.

8. Remove Unnamed zero and it’s dropping from dfs like Optical.csv

9. No null values after mosaic. (if cloudmasking was done you’ll have)

10. It would be useful to represent satellite data as a 3d numpy array. Not just for time series ML, but also for preprocessing like Cloudfilling.

11. In EDA, you perform differences/similarities b/w samples of the same region.

12. Definitely making a lot of assumptions about the data. Really would like to sanity check everything, down to the last minute details. Rows, columns, ranges etc.

13. Timing executions.

14. The way I wrote the code so far, It would take edits in several places to add arguments to a function. Maybe consider sending parameters with asterik in pipeline\_executable. Ofcourse, this doesn’t make sense when pipeline\_executable is calling multiple functions.

15. Up-to-date with SAR.

16. Percentage features.

17. Copying of hm rows into satellite data df is happening manually. This needs to be a util function with parameters that decide on which ones need to be copied. However, sometimes its not easy, because its rowwise (see gdd script)

18. Possible mistake – check if 2023 10 11 has clouds anywhere.

19. Check if some columns are completely empty. This wouldn’t make sense at all.

20. Basic checks like the function atleast changed the dataframe or other intended behaviours.

21. Fix column syntaxes only needs to be in one place. All others should check and raise error.

Loaded in github until this. (Future work part)

++++++++++++++

#################################################################3333333

The GDD error is quite weird. It is also related to the cloud error. Check and solve both.