Model documentation and write-up (3rd place)
Name: Kudaibergen Abutalip
Hometown: Nur-Sultan, Kazakhstan

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.**

My name is Kudaibergen Abutalip. Currently, I am a Master's student in Computer Vision at the Mohamed bin Zayed University of Artificial Intelligence and residing in Abu-Dhabi, UAE. Originally, I am from Kazakhstan.

2. **What motivated you to compete in this challenge?**

In addition to the scale of the problem being tackled and its importance, introducing myself to a new domain and getting much more familiar with new concepts seemed exciting for me. Additionally, I was interested in working with geospatial data.

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

The solution focuses on fusing approved Aerosol Optical Depth (AOD) data [1] and meteorological data [2]. Several research papers have indicated the importance of complementing AOD data with such features [3, 4, 5]. The final model was an ensemble of tuned Random Forest Regressor and generalized Gradient Boosting Regressor from the sklearn library.

3.1 Data Preprocessing

AOD data is in HDF format and appropriate preprocessing steps outlined by organizers were followed: layer extraction, adding offset and scaling, constructing the grid in WGS84(EPSG:4326) coordinate system. As the data in a raster format, mean, 95th percentile, min, max, standard deviation, variance of AOD values were extracted. The main problem with AOD data were missing values. Several approaches for time series imputation were tested among which linear interpolation performed the best. Data was interpolated for each grid id separately while ensuring proper placement of data points (sorting by date).

GFS data variables were selected based on some investigation. For example, authors of [6] indicate that meteorological factors might considerably affect PM2.5 concentration levels, and summarize correlation levels for rainfall, precipitation, wind speed, humidity, wind direction, atmospheric stability, relative humidity, daily average temperature, minimum temperature, maximum temperature. More specific descriptions are available in data -> raw -> gfs -> metadata_selected_variables and gfs_ds084.1_meta_selected.csv. GFS data is distributed in grib2 format at 0.25° and was aggregated with respect to the product specifications. Grid coordinates at a finer resolution of each grid in three locations were rounded to the nearest GFS grid to join with AOD values.

Feature engineering steps, which made into the final pipeline, involved inclusion of deriving year, month, and day features from the datetime column, wind magnitude from u and v directions of wind, mean encoding, label encoding, and addition of geospatial coordinates from grid metadata.

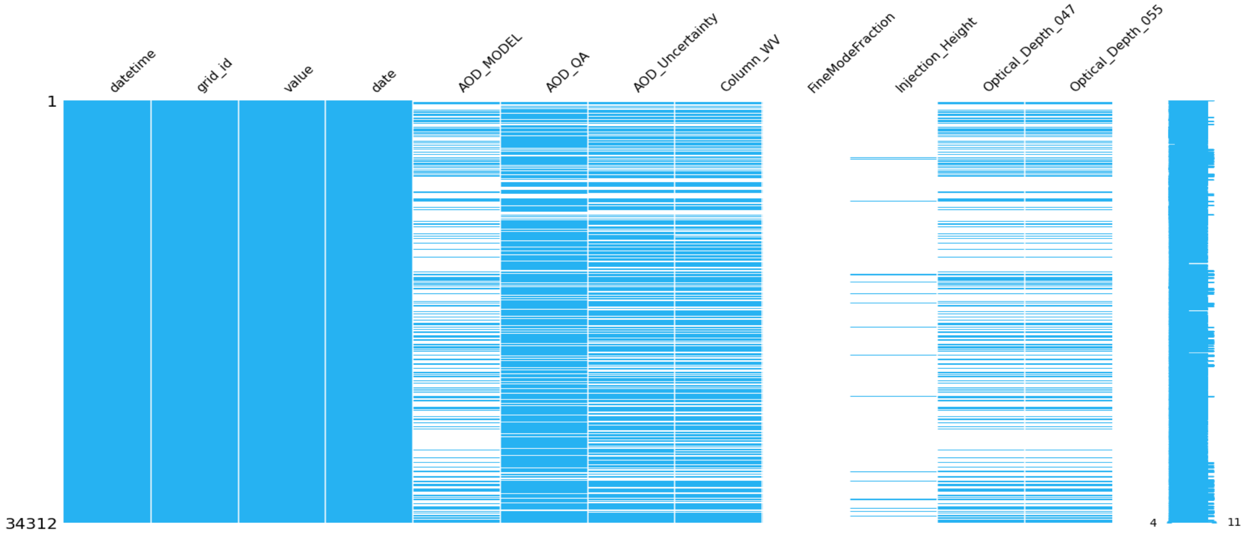
3.2 Models

The final model was an ensemble of tuned Random Forest Regressor and generalized Gradient Boosting Regressor from sklearn library. The former model was tuned using 'Optuna' hyperparameter tuning framework (example in nbs/modelling_optuna.ipynb), while default hyperprameters without fixing the random seed for the latter were used. Some experimentation showed such an approach more effective for increasing overall generalization perfomance of the pipeline. Saved models trained on full data are saved in ‘models’ folder. Corresponding names are ‘rf_winning_02.04_joblib’ and ‘grb_winning_02.04_joblib’.

models/predictions/subm_16.03_rf(tuned)_gr_ens_mean_windmag_full_data is the final 3rd place winning submission file

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

The following visualization shows entries with missing values in white color.



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

Some simple solutions which helped a lot:

```
1) imputing_method = 'interp_linear'
   df_filled = fillna_by_grid(df_p, imputing_method)
   df_filled.dropna(inplace=True)
```

Interpolation of the values instead of dropping them considerably, as expected, helped the model.

```
2) df_filled['wind_magnitude'] = np.sqrt(df_filled.u ** 2 + df_filled.v ** 2)
```

Feature engineering plays a crucial role in such tasks. Having more domain knowledge would help to generate more of these kind of new attributes.

```
3) avg_val = temp_df[var_of_interest].mean()
   min_v = temp_df[var_of_interest].min()
   max_v = temp_df[var_of_interest].max()
   variance = temp_df[var_of_interest].var()
   std = temp_df[var_of_interest].std()
  percentile_95 = temp_df[var_of_interest].quantile(0.95)
```

Computing other statistical measures of AOD values sufficiently helped the model to formulate the relationship with target variable.

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

The solution, originally, was run on Windows 10 Pro (Version 10.0.19044 Build 19044)
- Number of CPUs: 4
- Processor: Intel® Core™ i5-9300H 2.4-4.1 Ghz
- Memory: 16 GB

Data preparation time: For training data (35.1Gb): ~12-15 hours (no multiprocessing), 4-6 hours (with multiprocessing)
For testing data (20.3Gb): ~8-12 hours (no multiprocessing), 3-4 hours (with multiprocessing)

Training time: ~3 minutes

Inference time: ~2 minutes

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.)?

A problem with using multiprocessing with pyproj package and initializing EPSG:4326 coordinate system. For some reason, it might not work with multiprocessing in some environments or on some machines. I have tested it on both Windows and Linux with different conda distributions, python versions, and have not found a reasonable solution for this. During the competition period, it worked smoothly and the issue was identified only recently. Fortunately, the loop version of the function is available, which worked properly in all experiments.

Upon receiving GFS data, you might receive variables with unknow names. For group 2, unknown one is categorical rain, and for group 1, unknown one is sunshine duration. In the first case, they send only 2 variables and values are binary. In the second case, values are in seconds (e.g. 21600), which means temperature (measured in K) or humidity (widely-known definitions and example indicate that it can not be possibly that large[7]) are excluded from possible options. Besides, some analysis reveals that this variable has a strong positive correlation with seasonal climatic changes.

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

I only used HDFView application (https://www.hdfgroup.org/downloads/hdfview/) for getting familiar with HDF format. But it wasn't involved in the overall pipeline for perprocessing.

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

During training, I have used Time-Series Cross-Validation split with 4 folds and was using the same metrics (RMSE, R^2). Also, training-validation split was used for evaluating missing data imputation techniques and overall predictive performance.

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

Due to time constraints, deep learning models weren't investigated exhaustive enough. Though the number of recordings is small, I tried some new approaches, such as Temporal Fusion Transformer, or more known ones, like LSTM or MLP. However, initial experiments did not outperform the tree-based models. I assume, more investigation is needed to make these models work on this dataset. I also experimented with LightGBM, and XGBoost variations of tree-based boosting models. They were much more prone to overfitting and less constrained versions didn't outperform the version from sklearn.

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?

As I have mentioned in the previous answer, I would try to dig more into deep learning methods for both prediction and missing data imputation. The problem is, Aerosol Optical Depth values are missing in the MAIAC dataset. I would also investigate MISR data more. Another interesting aspect to look more into would be feature engineering. Acquiring more domain knowledge would help to come up with more useful features. Also, I wasn't able to successfully incorporate lag, and rolling features with missing dates.

References:
[1] MCD19A2 product: https://cmr.earthdata.nasa.gov/search/concepts/C1000000505-LPDAAC_ECS.html

[2] GFS Forecasts products: https://rda.ucar.edu/datasets/ds084.1/#metadata/detailed.html?_do=y

[3] Chu, Y.; Liu, Y.; Li, X.; Liu, Z.; Lu, H.; Lu, Y.; Mao, Z.; Chen, X.; Li, N.; Ren, M.; Liu, F.; Tian, L.; Zhu, Z.; Xiang, H. A Review on Predicting Ground PM2.5 Concentration Using Satellite Aerosol Optical Depth. Atmosphere 2016, 7, 129. https://doi.org/10.3390/atmos7100129

[4] Zhang, G.; Rui, X.; Fan, Y. Critical Review of Methods to Estimate PM2.5 Concentrations within Specified Research Region. ISPRS Int. J. Geo-Inf. 2018, 7, 368. https://doi.org/10.3390/ijgi7090368

[5] Karimian, H., Li, Q., Wu, C., Qi, Y., Mo, Y., Chen, G., Zhang, X. and Sachdeva, S. (2019). Evaluation of Different Machine Learning Approaches to Forecasting PM2.5 Mass Concentrations. Aerosol Air Qual. Res. 19: 1400-1410. https://doi.org/10.4209/aaqr.2018.12.0450

[6] Lin Y, Zou J, Yang W, Li CQ. A Review of Recent Advances in Research on PM2.5 in China. Int J Environ Res Public Health. 2018 Mar 2;15(3):438. doi: 10.3390/ijerph15030438. PMID: 29498704; PMCID: PMC5876983.

[7] https://www.zehnderamerica.com/absolute-vs-relative-humidity-whats-the-difference/#:~:text=Absolute%20humidity%20is%20the%20measure,water%20vapor%20%E2%80%93%2030g%2Fm3.