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.

   I'm a freelance data scientist.

2. What motivated you to compete in this challenge?

   Curiosity. I'm continually amazed by the power of remote sensing in impacting our lives on earth. Having participated in several satellite data competitions, I've come to appreciate the immense value that can be derived from this information, and its ability to reveal previously unseen insights about our planet.

3. High level summary of your approach: what did you do and why (e.g., key features, algorithms you used, any unique or novel aspects of your solution, etc.)?

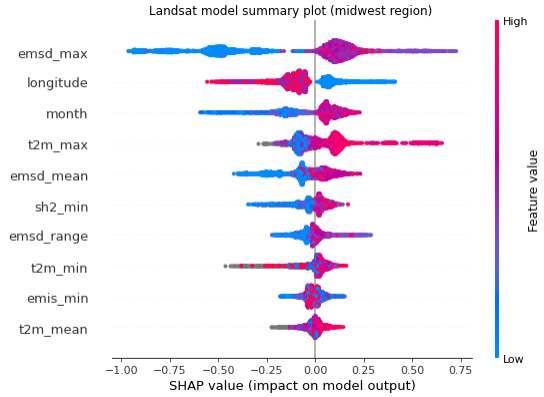
   My solution uses level 2 data from Landsat 8, 9, and Sentinel 2. For each sample, I extracted observations within a 200m radius of the sample location and up to 15 days prior to the sample collection date. Additionally, I incorporated temperature and humidity HRRR forecasts from the past 24 hours relative to 1200 UTC of the sample collection date.

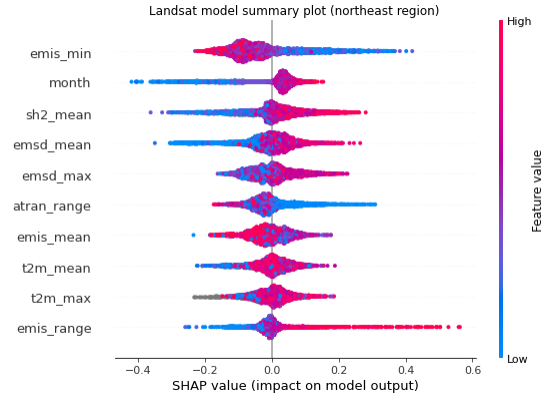
   To tackle the problem, I formulated it as a regression task and employed a gradient boosting algorithm to directly forecast the cyanobacteria severity level. To ensure robustness, I divided the dataset into five stratified folds and trained a separate model for each satellite data and region. The final prediction for the test data is a mean ensemble of predictions from 10 models (2 datasets x 5 folds), rounded to the nearest severity level. In cases where the data was unavailable, I set the severity level to the average of the region's predictions.

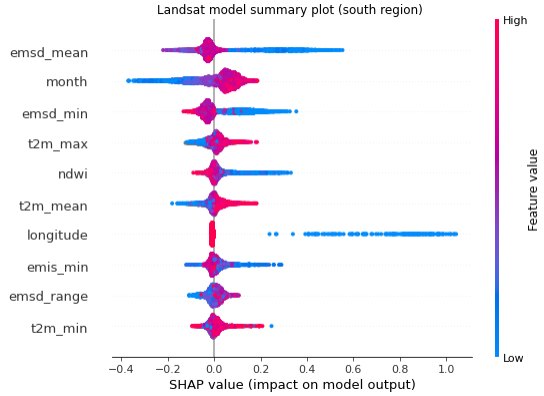
4. Do you have any useful tables, charts, graphs, or visualizations from the process (e.g., summarizing model performance, testing different features, exploring the data, etc.)?

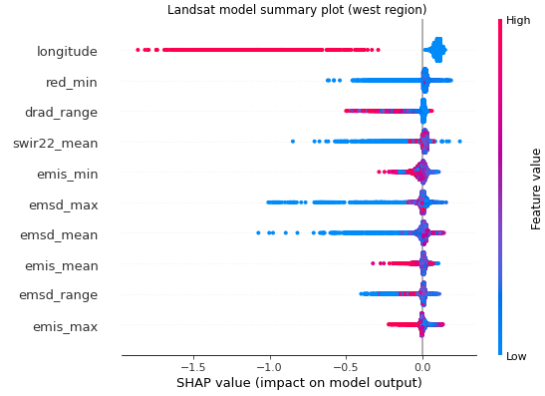
Feature importance - SHAP summary plots

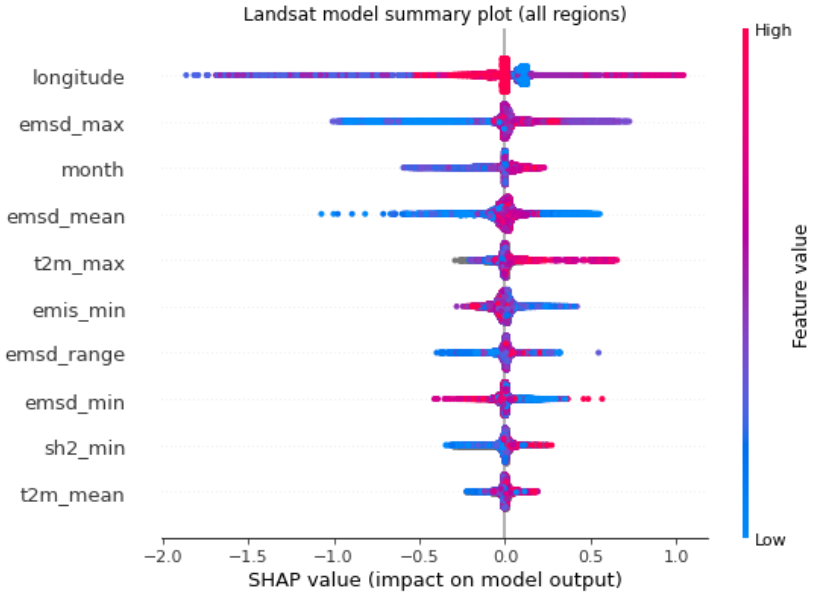
Landsat dataset models



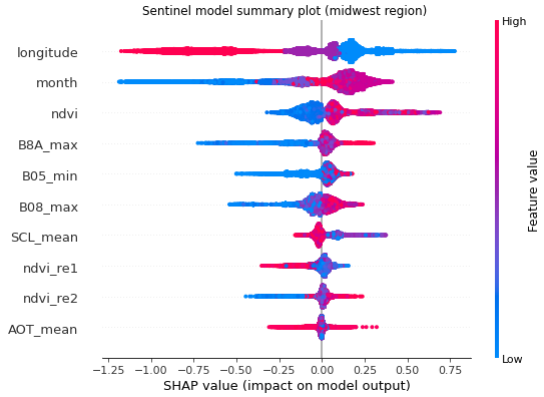


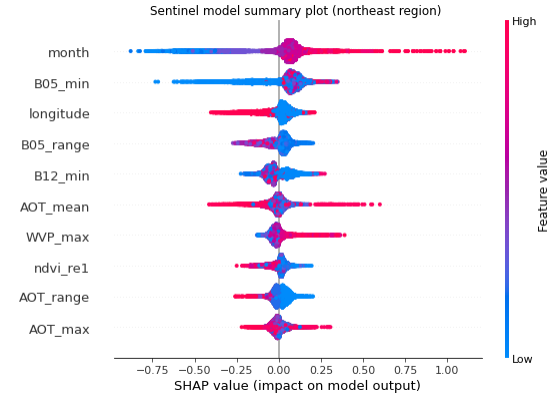


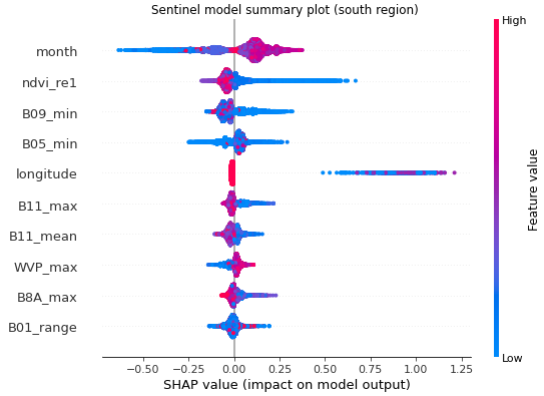


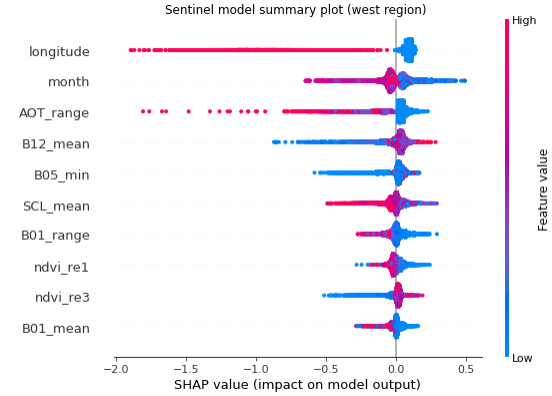


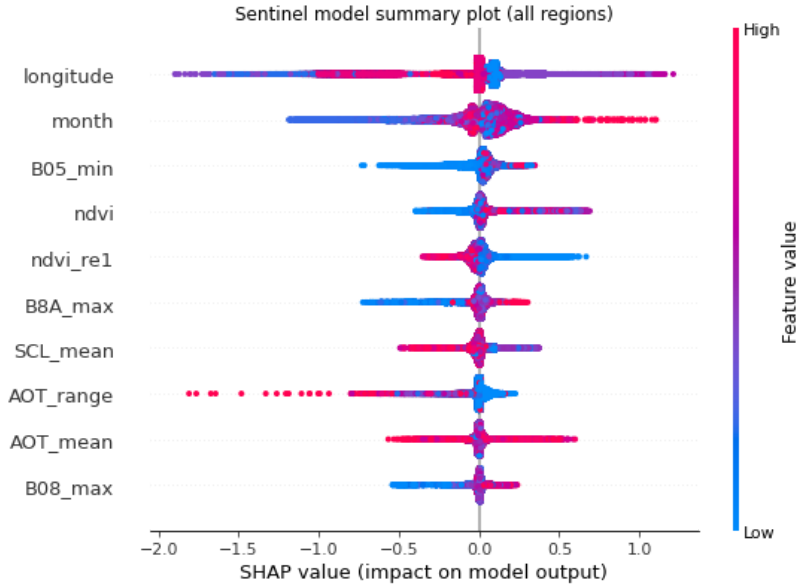
Sentinel dataset models











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

   Some samples returned by the moe

```
def calc_sample_quality(path):

  x=np.load(path)['arr_0']

  if x.mean()<1:

    return 0

  return 1
```

   Sorting by quality, cloud cover and date to pick out the most informative samples
   I kept only the first sample for landsat data and the top 15 samples for the sentinel data.

```
data=data.sort_values(by=['quality','cloud_cover','datetime'],ascend
ing=[False,True,False]).groupby('uid').head(nsamples)
```

6. Please provide the machine specs and time you used to run your model.
   - CPU (model): Intel Xeon @2.20 GHz
   - GPU (model or N/A): N/A
   - Memory (GB): 12
   - OS: Ubuntu 21.04
   - Train duration: <10 minutes
   - Inference duration: <10 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, failure modes, etc.)? NA

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

9. What features did you select for use in your model and why? Please describe the preprocessing steps you took to develop features from raw data (e.g., filling missing values, removing outlier values, encoding categorical values, etc.), and what criteria or methods you used to select these features.

   I began by generating all the features that I could think of from the available data:

   - Statistics of the raw satellite observations from up to 15 days preceding the sample date - mean, max, min, range
   - Remote sensing indices calculated from the mean statistics of the satellite data
   - Statistics of HRRR forecasts from 24 hours before the sample date @1200 UTC - mean, max, min
   - Metadata - time (sample month, days between a satellite observation and the sample date), longitude
   - Elevation statistics - mean, max, range, skewness

   Next I calculated the drift of each feature across the training and test datasets with a simple binary classifier and selected features with a low drift score (AUC<0.8).

   Finally, I sequentially added each feature to the model and kept them if they improved performance in both local validation and the public leaderboard. I ended up with a total of 155 features.

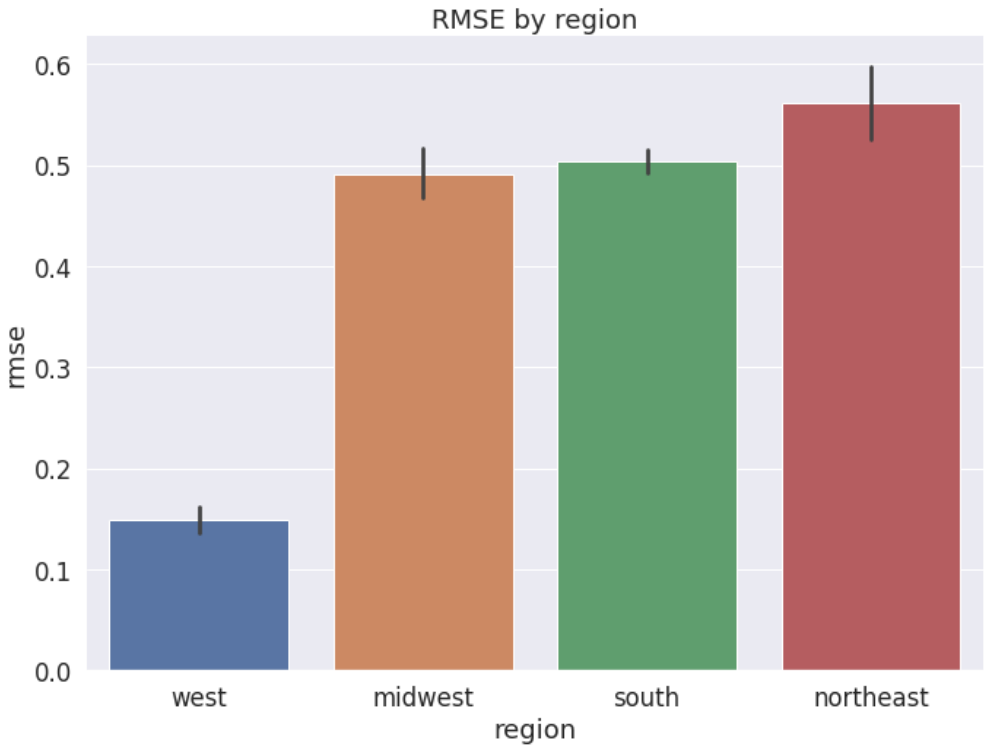
   The following table shows the most significant features based on feature importance analysis.

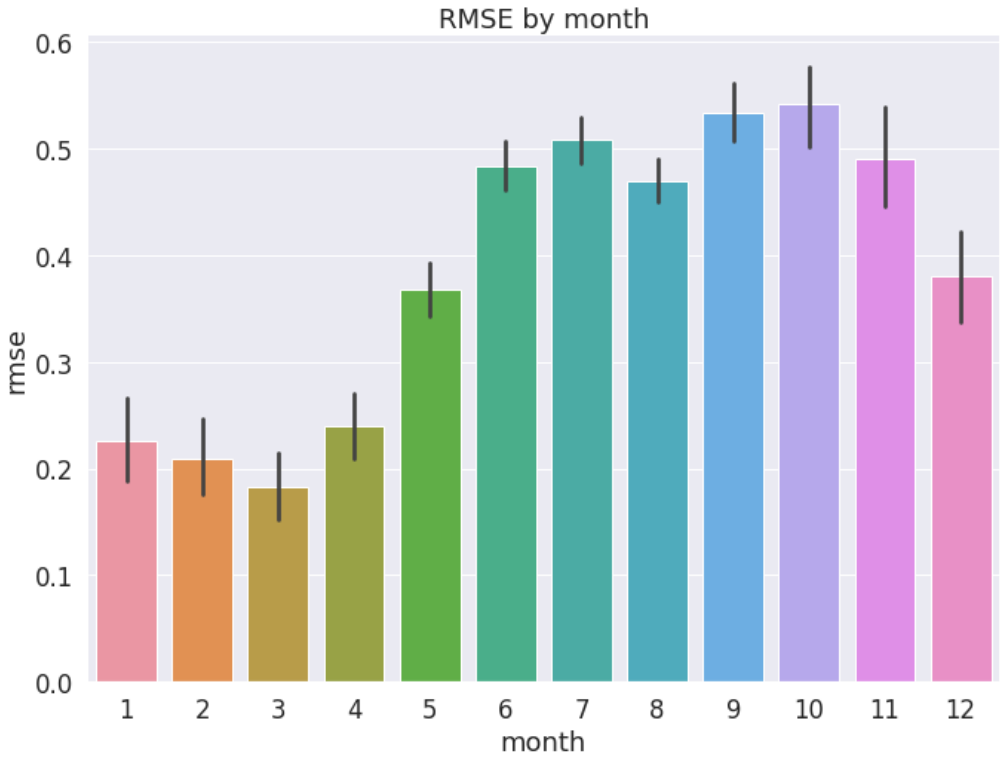
Data source	Feature	Description
Landsat 8,9	atran	Atmospheric transmittance
	drad	Downwelled radiance
	emis	Emissivity
	emsd	Emissivity standard deviation
	red	Surface reflectance band 4
	swir22	Surface reflectance band 7
Sentinel 2	B01	Coastal aerosol
	B05	Red edge 1
	B08	Near infrared (NIR) 1
	B8A	Narrow NIR
	B09	Water vapor
	B11	Short-wave infrared (SWIR) 1
	B12	SWIR 2

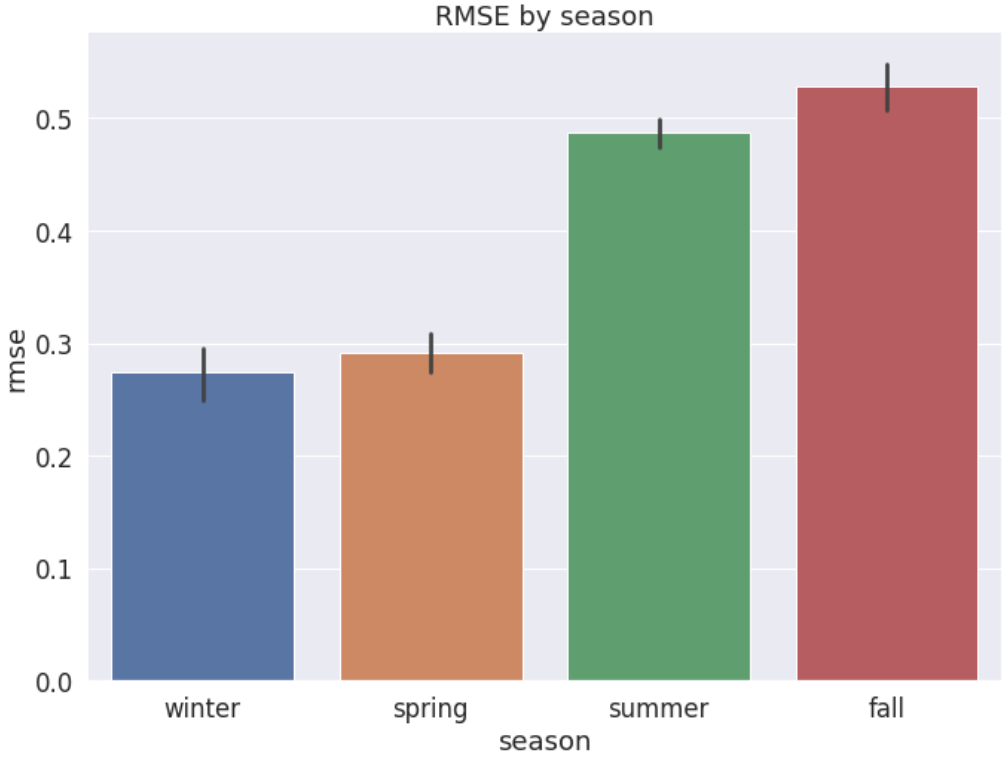
	AOT	Aerosol optical thickness
	SCL	Scene classification layer
	WVP	Scene average water vapor
	ndvi	Normalized Difference Vegetation Index
	ndvi_re1	Modified NDVI with Red edge 1
	ndvi_re2	Modified NDVI with Red edge 2
	ndvi_re2	Modified NDVI with Red edge 3
HRRR	t2m	temperature forecast at 2m above ground
	sh2	specific humidity forecast at 2m above ground
Metadata	longitude	sample collection location
	month	sample collection month

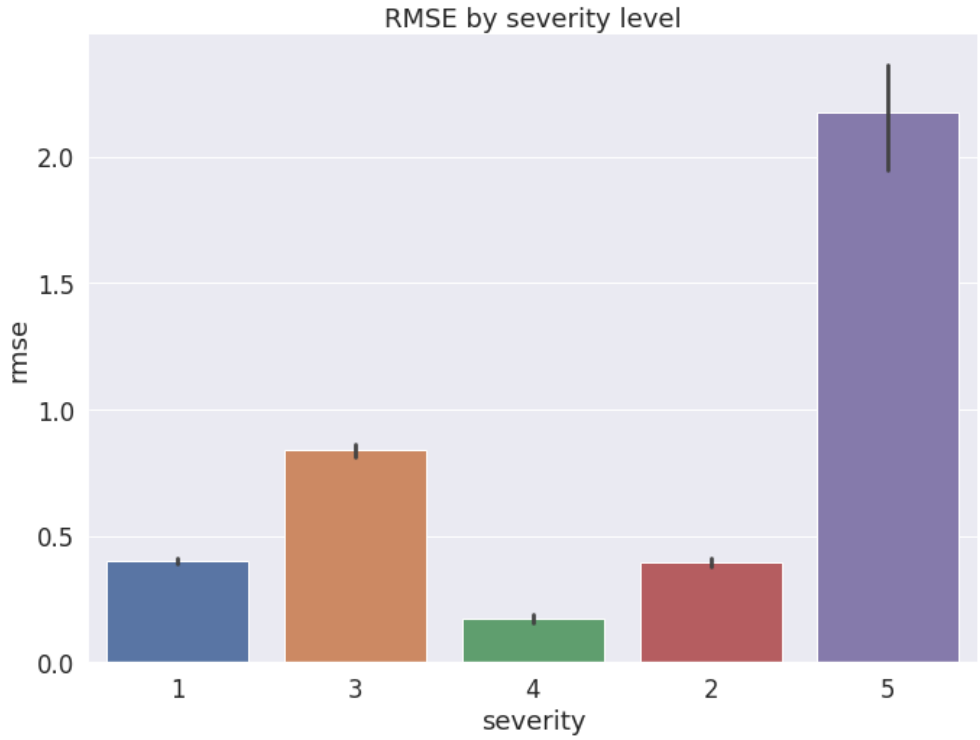
10. How does your model's performance vary? Please consider model performance for various levels of cyanobacteria density, region, environmental conditions, and season. Describe under which conditions your model is the most accurate and under which conditions your model is the least accurate.

    The following charts show RMSE scores of the out-of-fold predictions in various scenarios:




RMSE by month
rmse
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1
2
3
4
5
6
7
8
9
10
11
12
month


RMSE by season
rmse
0.0
0.1
0.2
0.3
0.4
0.5
winter
spring
summer
fall
season



11. How did you evaluate the performance of your model (e.g. how you approached validation, dataset splits, metrics used, etc.)?

    I monitored the training and validation weighted RMSE scores and stopped training if the scores didn't improve after 100 iterations.

12. What simplifications could be made to run your solution faster without sacrificing significant accuracy?

    Fairly simple model, speed shouldn't be an issue.

13. What are some other things you tried that didn't necessarily make it into the final workflow (e.g., features, preprocessing steps, etc.)? Did you test any other types of models?

    The HRRR forecasts did not perform as well as expected. I tested various variables that could impact cyanobacteria formation and movement, such as temperature, humidity, precipitation, and wind. Of these, only temperature and specific humidity improved the model's performance on both the local validation and test sets.

    Intuitively, elevation should also affect cyanobacteria formation due to its effect on runoff and water stagnation. However modeling with elevation data overfitted the training set and performed poorly on the test set. This suggests a complex relationship between elevation

and algae blooms that may be influenced by other factors, such as weather conditions and nutrient levels.

14. 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?

    - Filtering and mosaicking - filtering clouds and non-water pixels and combining the satellite observations for each sample
    - Exhaustively test performance of HRRR forecasts - other variables, multi-day instead of only the past 24 hours
    - Feature selection by region
    - Experiment with imputation
    - Parameter tuning, try other models

    The biggest challenge was correlating local validation with the test dataset. Despite eliminating features that exhibited significant drift between the train and test datasets, results varied unpredictably.

Write-up rubric

Write-up submissions will be evaluated along the following dimensions for the bonus prize for best write-ups.

Interpretability (35%): How comprehensively does the report describe the full model pipeline from data intake to inference (e.g., feature engineering, model selection, training and validation procedures)?

Insight (25%): How well does the report describe the exploratory, trial-and-error process that led to the given solution (i.e., the rationale for what was tried and what was ultimately used)?

Rigor (25%): To what extent is the report built on sound, sophisticated quantitative analysis and a performant statistical model?

Clarity (15%): How understandable is the report for a broader audience (i.e., a subject matter expert without technical experience in python-based ML)?