Model documentation and write-up

1. Who are you (mini-bio) and what do you do professionally?

I write and self-publish books on machine learning with a focus on topics that go beyond just training the models, such as uncertainty quantification and interpretability. I have a background in statistics and machine learning.

2. What motivated you to compete in this challenge?

I love to write, but sometimes I need practical projects. Especially when writing practical books, I don't want to lose touch with the practical side of machine learning. The Water Supply Forecasting Rodeo fit that perfectly and checked many other boxes too: I am interested in all things earth system science, and the challenge had an additional bonus track on explainability and communication, as well as attractive prizes.

3. If there are any particularly useful snippets of code when producing your communication outputs (calculations, visualizations, etc.) that would be useful to highlight, please copy-paste them here.

When it comes to interpreting machine learning models, many approaches are model-agnostic, meaning they can be applied to any model. You can use model-agnostic methods to explain predictions from linear regression models, but also for an ensemble of 10 XGBoost models (which I used for each quantile). The methods I used were Shapley values and What-if plots and both are model-agnostic.

For the explainability and communication outputs, we should not only explain the forecasts but also the uncertainty, measured as the range between the 10% and 90% quantile. Since I used model-agnostic methods, I could use the same methods by simply redefining what the "model" output is. To explain the uncertainty, I defined the interval range as the "prediction function":

```
def predict_interval(X):
        return predict_wrapper(X, 0.9) - predict_wrapper(X, 0.1)
```

Otherwise, I could use the same algorithm (Shapley values and What-if plots) to generate the communication outputs.

4. Please provide the machine specs and time you used to train your model and to produce the communication outputs.

- CPU (model): Apple M1
- GPU (model or N/A): Apple M1
- Memory (GB): 16 GB
- OS: macOS 14
- Train duration: ~2 min (excluding data downloads)

- Inference duration: <1min
- Forecast report creation: ~40 seconds per report

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

I generate the forecast reports using Quarto. The report template file (forecast.qmd) is written in Quarto Markdown (.qmd), which is a flavor of Markdown. The file contains code chunks with both Python and R and passes objects from Python to R. The forecast.qmd may look a bit unusual if you are seeing such a file for the first time.

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

Creating explanations for forecasts is more difficult when features are strongly correlated because then it becomes difficult to attribute the effects to each of the correlated features. Methods like Shapley values produce explanations by creating new data points that often ignore the correlation and then produce unrealistic data for the explanations. To counteract this, I analyzed the correlations beforehand to decide which highly correlated features to group (SWANN and SNOTEL SWE estimates were grouped).

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

I didn't try anything other than Shapley values and What-if plots, but I did try many different ways of visualizing them. Getting all the explainability and communication outputs to fit into just 2 pages was challenging and involved a lot of trial and error.

8. If you were to continue working on your explainability/communication solution for the next year, what methods or techniques might you try in order to build on your work so far? Are there other metrics or visualizations you felt would have been very helpful to have?

Before I would improve anything, I would talk to the hydrologists and decision makers who use these outputs. Are the visualizations and outputs easy to understand?

Other than that, I would:

- Automate the narrative analysis. It should be possible at least in part.
- The correlations between features varied from site to site. To improve the explanations, it might make sense to have different feature groups for different sites.