Snow and Flow: Water Supply Quantile Regression with Few Features

Christoph Molnar (kurisu)

This report presents a water supply forecasting solution based on quantile regression using gradient-boosted trees (xgboost). The models are trained for all sites and issue dates simultaneously. To predict each quantile, an ensemble of 10 xgboost models is used in combination with adjusting lower and upper quantiles to ensure an 80% interval coverage. Besides issue date and site, the models rely only on features describing snow conditions (SWANN, SNOTEL) and antecedent flow (NRCS, RFCs). Training takes 6 seconds, prediction is even faster, the feature list is short and interpretable, and the models achieve site-wise coverage.

1 Technical Approach

1.1 Algorithm and Architecture Selection

At the start of the competition, I chose to use quantile regression with gradient-boosted trees. There are other methods available, such as Bayesian models, regression with uncertainty quantification, and bootstrapping. But quantile regression is the most straightforward approach for minimizing the average mean pinball loss. I used the gradient-boosting framework xgboost (Chen and Guestrin 2016), which is a solid choice for modeling tasks with tabular data (Shwartz-Ziv and Armon 2022). I experimented with other quantile regression algorithms, such as LightGBM, CatBoost, and linear quantile regression, but they weren't as performant and slower (LightGBM, CatBoost).

> **i** xgboost
>
> xgboost is a machine learning algorithm that trains decision trees sequentially, with each tree correcting the errors of the previous one.

This solution consists of an ensemble of 10 xgboost models per quantile, each of which optimizes the respective quantile loss for the 10%, 50%, or 90% quantile. To make a prediction, follow these steps: 1) create the features, 2) for each quantile, get the 10 predictions from the respective ensemble; 3) average the predictions per ensemble, 4) if the issue date is in-season add the flow of the previous in-season months, 5) adjust the lower 10% and 90% quantiles, and finally 6) post-process the quantiles. Figure 1 visualizes these steps.

The process of adjusting and post-processing of the quantiles is explained in Section 1.3. The modeling solution runs efficiently by relying on only 7 features and making predictions for all 26

sites simultaneously. This means the models had to learn to differentiate between dates and sites through features. I tried to train separate models for each issue date or even for each issue date and site, but the performance was slightly worse.

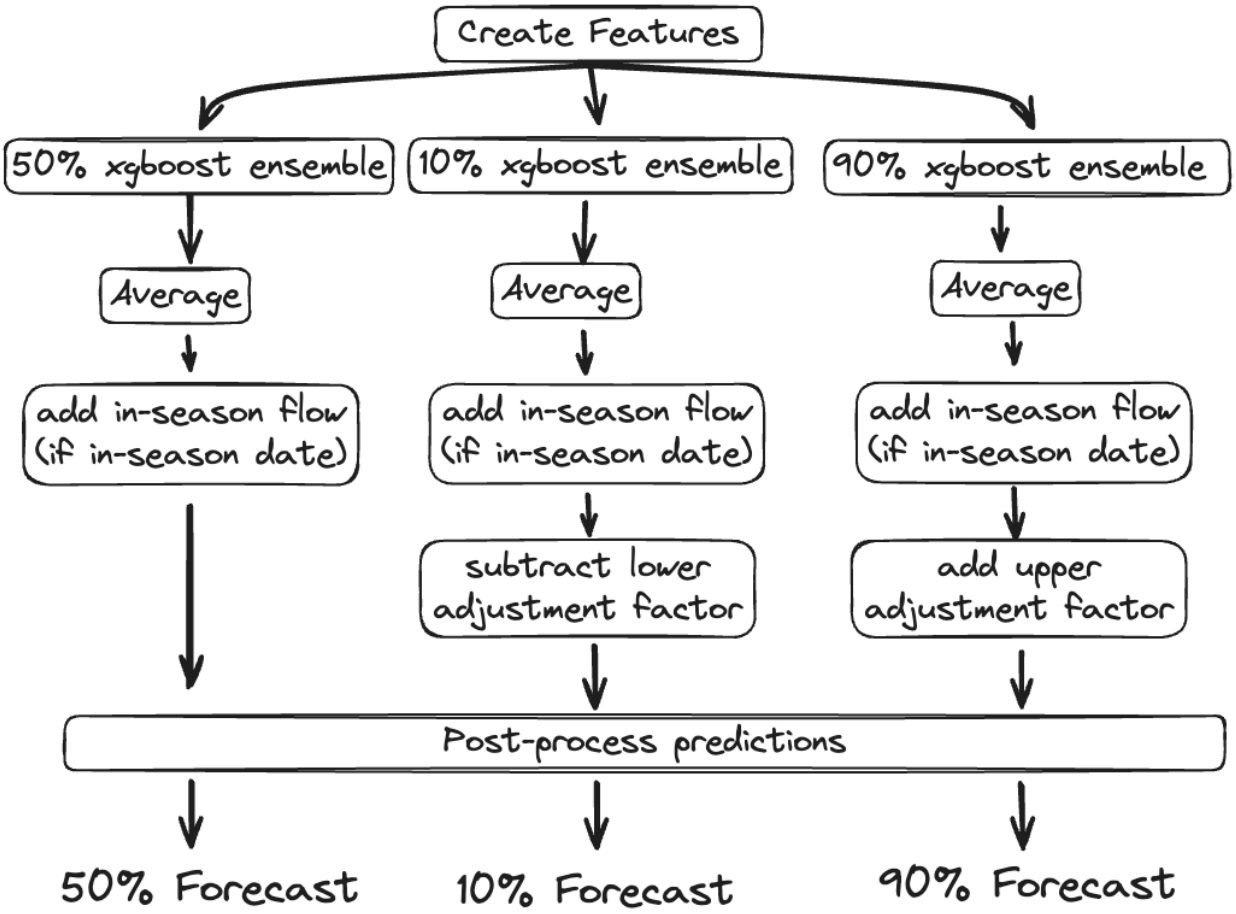

Figure 1: Steps to make a prediction

1.2 Data Sources and Feature Engineering

1.2.1 Data sources

Before the competition I knew very little about hydrology and hustled through the list of approved data sources. Eventually, I discovered that the snow condition (via SNOTEL) is the main predictor that also previous approaches used (Fleming and Goodbody 2019; "Statistical Techniques Used in the VIPER Water Supply Forecasting Software" 2023). During the forecast stage I discovered that including SWANN estimates boosts model performance. The final stage modeling approach uses the following data sources:

- **Antecedent streamflow**: NRCS and RFCs monthly naturalized flow
- **Snowpack and precipitation**: NRCS SNOTEL + UA SWANN

I considered other approved data sources as well, but none of them improved predictive performance.[1]

1.2.2 Feature engineering

Table 1 describes the features used in all quantile models:

Table 1: Features used

Feature	Source	Comment
site_encoded		Categorical feature differentiating sites
day_in_year		Days since beginning of calendar year
antecedent_flow	NRCS/RFCs	A site's flow from previous month
snotel_swe_conditional	SNOTEL	Latest relevant snow-water equivalent
swann_swe_conditional	SWANN	Latest relevant snow-water equivalent
swann_ppt_conditional	SWANN	Latest relevant accumulated precipitation
swann_ppt_unaccounted	SWANN	In-season unaccounted precipitation

The first three features can be quickly explained as follows:

- **site_encoded**: This is a categorical feature that encodes the 26 sites and enables the models to differentiate between them.
- **day_in_year**: This is a numerical representation of the day in the calendar year, ranging from 1 for January 1st to 204 (in a leap year) for July 22nd.[2].
- **antecedent_flow**: This feature is the site's naturalized flow measured the month before.

Computing SNOTEL features

To compute the SNOTEL feature snotel_swe_conditional, follow these steps (visualized in Figure 2):

1. Select "SNOTEL stations within approximately 40 miles of the forecast site drainage basins" based on the code kindly provided by the competition organizers.
2. Calculate daily averages of the snow water equivalent variable (WTEQ_DAILY).
3. Conditional on the issue date, select a previous daily average.
4. If no SNOTEL data is available, return a missing value.

The process for selecting the previous daily average conditional on the issue date is explained and visualized in Figure 5 later in this section.

Computing SWANN features

The SWANN data product provides data in two formats: spatially averaged based on Hydrologic Unit Codes (HUC) and 4km gridded data. For this solution, I opted for the HUC-based approach.

[1]Datasets I tried: USGS streamflow, RCC-ACIS, CPC seasonal outlooks, GRACE-based Soil Moisture and Groundwater Drought Indicators, all the teleconnection data with their 3-letter acronyms: ONI, SOI, MJO, PNA, and PDO, and SNODAS.

[2]Training decision trees, which form the basis of xgboost models, is insensitive to constant shits in a feature. Therefore, changing the day-in-year feature to "days in water year" would produce a model with identical predictions.

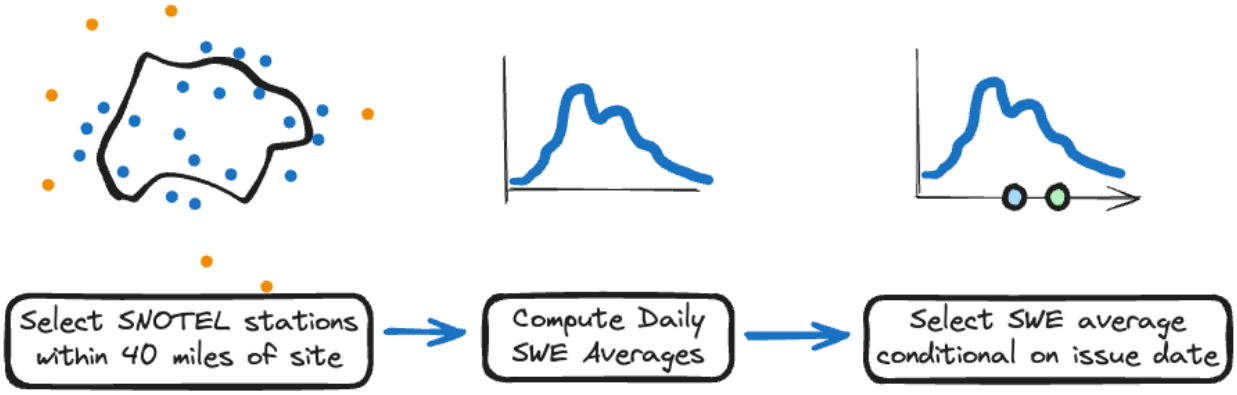

Figure 2: How SNOTEL data is aggregated

However, the HUCs used in the competition differ from those provided by SWANN. To address this, I used the 250k HUC dataset from USGS ScienceBase to establish a mapping between the competition sites and the corresponding SWANN HUC data.

The mapping process starts with overlaying the 250k HUC dataset polygons with the site basin polygons, identifying all HUCs with overlaps. I downloaded the respective HUC-based daily estimates and averaged the variables "Average SWE (in)" and "Average Accumulated Water Year PPT (in)" by day and site. Then I computed features 'swann_swe_conditional' and 'swann_ppt_conditional' by picking a previous date conditional on the issue date (again, explained in Figure 5). I calculated the 'swann_ppt_unaccounted' by subtracting the accumulated precipitation on the conditional date from the latest accumulated precipitation.

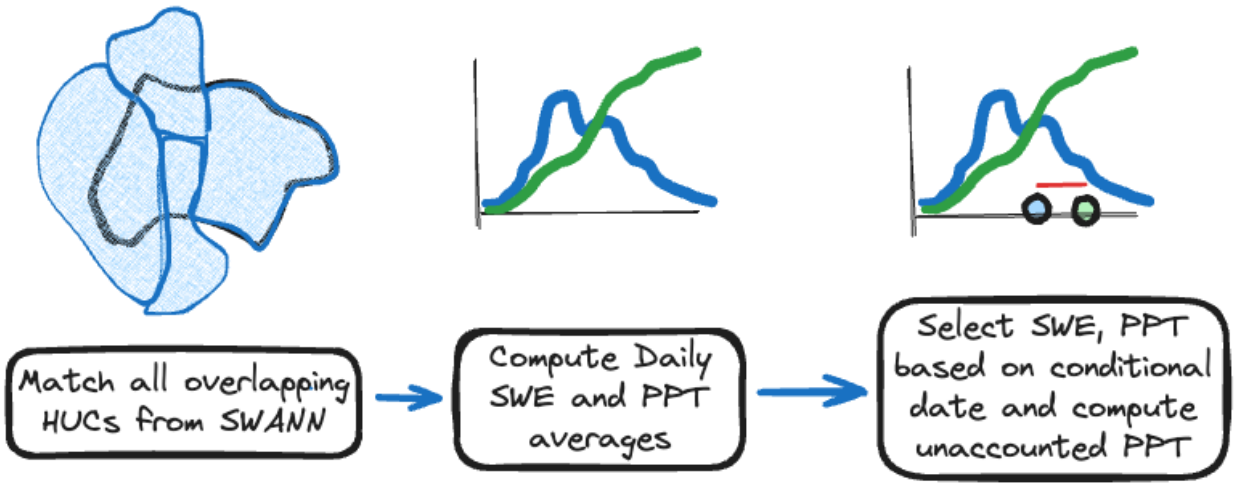

Figure 3: How SWANN data is aggregated

Fixing the "End-of-Month Melt Bias" with a conditional date

In hindcast and forecast stages I used the most recent SWE and PPT averages from SWANN and SNOTEL as features. However, the use of the latest snow data introduces a bias, which I call end-of-month melt bias. The bias becomes obvious when plotting the performance (average pinball loss) by site and issue date, as shown in Figure 4. Normally, the pinball loss should decrease

throughout the water year as more information about snow and flow become available. But due to the end-of-month melt bias, the loss increases during in-season months (from 1st to 22nd). This bias is most severe in May, but also occurs in April, June, and July. The pre-season months are not affected.

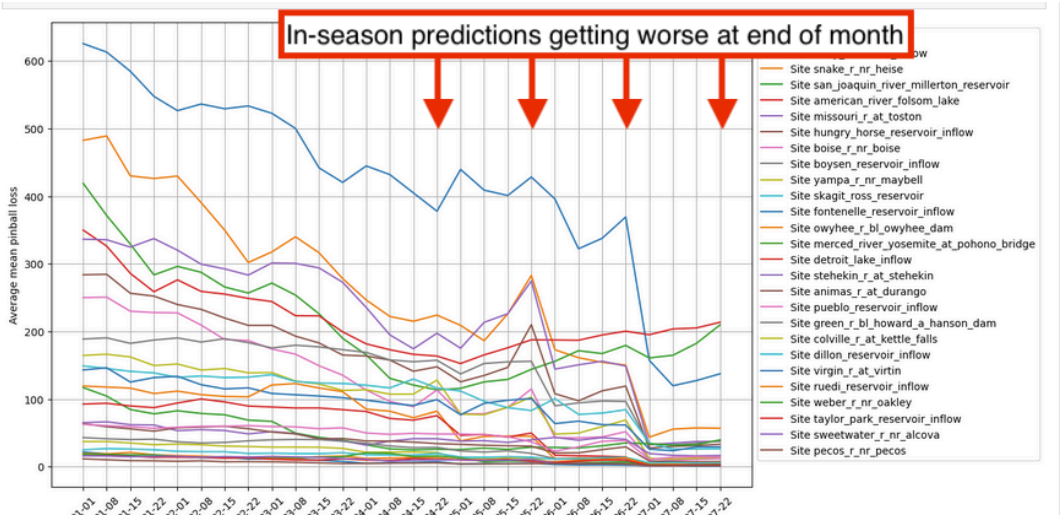

Figure 4: The "end-of-month"-bias for my previous model solutions: Predictions for in-season months get worse toward the end of the month. Each line represents the average mean pinball loss per site over the years.

Let's illustrate the end-of-month bias with an example: It's May 1st 2024 at Snake Heise and the mountain ranges are packed with snow. A water supply forecast model based on the latest snow estimates will use data from April 30th to make a prediction. By May 22nd, a good chunk of that snow has melted, which is reflected in the snow estimates from 21st of May. However, since it's still May, the model has no access to the monthly flow data from May. Information-wise, the model is worse of than on May 1st! This information gap is what I call the end-of-month melt bias. The bias is even worse for sites without monthly flow data, leading to a decrease in performance during the entire season.

To combat the end-of-month melt bias, I changed the date for which to retrieve the snow status: Instead of using the latest value, I use the last day of the month before (if issue date in-season) and in the case of sites without flow data, the retrieval even goes back to last pre-season day. This concerns features snotel_swe_conditional, swann_swe_conditional, and swann_ppt_conditional. Figure 5 illustrates how the snow status date is chosen based on the issue date and site. This removes the bias, but also means the models wouldn't get in-season updates on snow and precipitation (except when new monthly flow data becomes available). To close this information gap, I introduced the feature swann_ppt_unaccounted which measures the additional precipitation after the conditional date based on SWANN data. For pre-season months, swann_ppt_unaccounted is always 0.

Physical Intuition

This is my mental model of how the provided water supply forecasting operates:

- If the models only relied on "issue_date" and "site_id", they would model the historical quantiles per site and issue date.

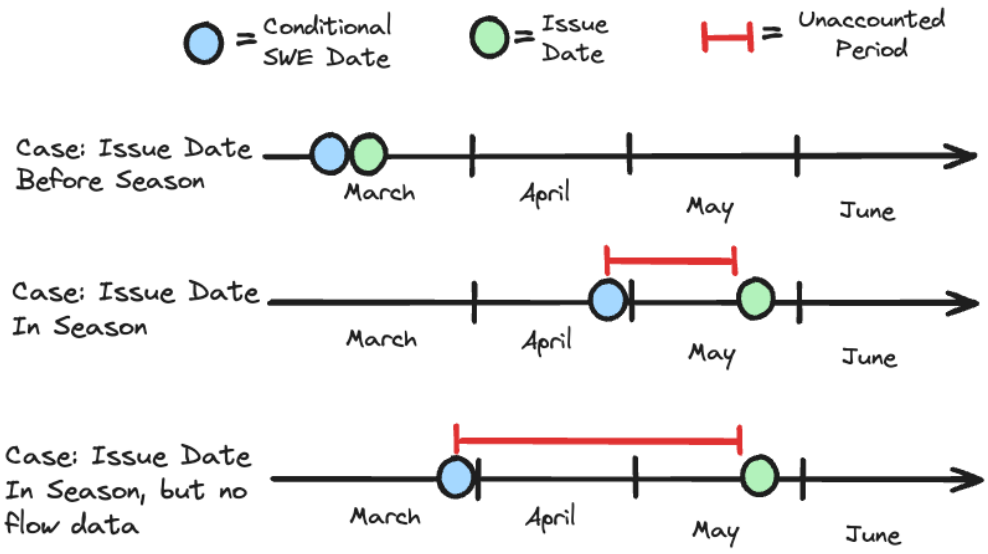

Figure 5: How conditional SWE dates are computed

- The most important information is the snow situation in the basin, which is captured by SWANN and SNOTEL.
- Speculation: The antecedent flow feature captures a lot of the remaining water year dynamic not covered in snow and precipitation features. That's why adding other features didn't improve the models.
- Relying on SWANN should make SNOTEL features obsolete, but having both in the model improves performance. Speculation: It's because SWANN ends at the US border, but water doesn't and some basins reach into Canada. SNOTEL data covers these areas. There might be other factors at play as well. For future models, it might be a good idea to merge the SWANN and SNOTEL features.
- Snowpack estimates include stations/areas outside the basins. While from a hydrological standpoint this out-of-basin snow doesn't drain into the basin of interest, my working theory is that casting a wider net stabilizes SWE and PPT estimates. Removing out-of-basin stations or using weighted averages of HUC regions worsened forecast performance.
- The prediction task changes between pre-season and in-season, both in the target (predicting total versus remaining volume) and relation between total volume and current snow content. This modeling solution covers this situation through how features are constructed.

i Missing data

Antecedent flow is missing for 3 of the sites and some early years don't have SNOTEL/SWANN data, leaving missing values in the features. XGBoost handles missing features by automatically determining whether to assign them to the left or right branch of a decision tree during training, based on which split maximizes model performance.

1.3 Uncertainty Quantification

My solution to the uncertainty quantification aspect of the competition is quantile regression: Instead of fitting one model (ensemble), you fit 3, one for the median, one for the lower quantile (10%), one for the upper (90%).

The upside: It's the most direct approach to optimizing for the competition's objective without having to make assumptions about the distribution of volume. The downside: Quantile models can suffer from **quantile crossing** (e.g. if the estimate for the 10% quantile is larger than the one for the 50% quantile) and **undercoverage** (e.g. if the 10% to 90% intervals contain less than 80% of the true outcome on average).

To correct for undercoverage, I adjusted the lower and upper quantiles, using an adapted version of conformal quantile regression (Romano, Patterson, and Candes 2019). Conformal quantile regression is a method to "fix" coverage issues of quantile regression by adjusting the lower and upper estimates. Adjustment terms are computed using an "unseen" calibration dataset. Then for predicting new data points, this adjustment terms is subtracted from the lower quantile and added to the upper quantile to ensure the desired coverage. My approach deviates from classic conformal quantile regression: Instead of learning a single term for both quantiles, I learn quantile-specific adjustment factors (for 10% and 90%). Section 1.4 explains the estimation process as it is intertwined with the model fitting.

However, **marginal coverage alone is misleading; it's better to look at coverage by site.**

The leaderboard shows marginal coverage, which is the coverage averaged over all sites, years, and issue dates. A model with a marginal coverage of 80% is not automatically a good model, because 80% can be achieved by having half of the sites with coverage of 60% and the other half with 100% coverage. Optimizing the average mean pinball loss should prevent this described scenario by some degree, however there is still an opportunity to "cheaply" improve coverage. The pinball loss depends on the scale of the target. And the scale differs quite a lot between sites: The Libby Reservoir, for example, has a seasonal volumes up to 1000 times larger compared to the smallest sites in the competition (Sweetwater). This means you can sacrifice homeopathic amounts of performance from the small sites to achieve better coverage.

That's why I would argue for looking at **coverage by site** instead. So in my modeling solution I therefore used group-wise conformal prediction, which means that the adjustment factors are calculated by site (and by issue date).

1.4 Training and Evaluation Process

The training data consisted of approximately 32,000 rows, with each row representing a unique combination of issue date, year, and site. All years after (and including) 1975 were part of the dataset. For pre-season issue dates, the target is the total volume, and for in-season dates, it's the total volume minus the accumulated in-season flow which is later added back into to the predictions for seasonal volume predictions. The models within each LOOCV-loop were trained in the following way:

- Use data between 1975 and 2023 without the "omitted" year in the LOOCV loop
- Split this data into 10 parts grouped by the forecast year (10-fold CV)
- Loop through the 10 parts:

  - Train the model with the combined 9 parts (~36 years)
  - Predict the remaining part (~ 4 years). These are the out-of-fold predictions.

- Save the 10 models
- Concatenate all the out-of-fold predictions
- Based on the out-of-fold predictions, calculate the lower and upper adjustments per site and issue date:

  - Lower: $Q_{90}(f_{10}(x) - Y)$
  - Upper: $Q_{90}(Y - f_{90}(x))$

For the adjustments calculation, $f_{10}(x)$ and $f_{90}(x)$ are the water supply predictions for the 10% and 90% quantiles and Q_{90} is the 90% quantile of the difference between the target and the upper and lower quantiles estimates. Some intuition behind these adjustment factors: If the models had perfect coverage, both the lower and upper adjustment factors would be close to zero. This is because with perfect coverage, the $f_{10}(x) - Y$ would be negative for 90% of the samples and positive for the other 10%, so the Q_{90} of the differences would be between the differences closest to zero. In a case of undercoverage for a site due to, e.g. f_{10} being too large, the respective lower adjustment factor $Q_{90}(f_{10}(x) - Y)$ would yield a positive number. For the prediction we would subtract this number which widens the interval and counters the undercoverage.

The cross-validation was done by year so that a water year could only be in either training or calibration, but not both. I used the by year split strategy to prevent data leakage and to reflect the use case of generalizing to new years.

A note on tuning: The models are not tuned, but I used xgboost with default hyperparameter settings. This may seem a bit unusual, but since the final model is an ensemble, instability in the individual models of the ensemble can actually desirable (Breiman 1996). While 32k rows sounds like a lot, it's a deceptive number, since there are only 520 unique values which are not even independent of each other, so the data is scarce. Using untuned xgboost models has several advantages: more data can be used for model training, the training setup involves fewer steps and it makes for a good ensemble.

1.4.1 Validation

The final prize stage is quite unusual for a machine learning competition and runs the risk of overfitting the final leaderboard. There are only 520 unique volumes (20 years x 26 sites) to predict, and they aren't even independent as they are correlated per site and per year. Throughout the development, hindcast, and forecast stages all participants have made decisions based on data that are now used for evaluation. We even have direct access to the outcome to be predicted and there is no private leaderboard.

It's fair to assume that all participants overfit on the 20 LOOCV years.

Some more, some less. And with my final prize stage submission I aim to be in the "less overfitting" camp, even if it means losing a few places on the leaderboard. For the final prize stage, I've based my model validation on two pillars:

- Changes made for the final prize stage model must improve the rigor of the model as listed in Section 3.

- In addition, I set up an evaluation based on LOOCV with the 10 years from 1994 to 2003. With the 94-03-LOOCV I can ensure that the model really gets better throughout the stages. The 94-03-LOOCV evaluation for my final model is very close to the final prize leaderboard. See Table 2.

Table 2: Model performance based on LOOCV using the years from 1994 to 2003. The models get better in each stage.

	Hindcast Model	Forecast Model	Final Model
Pinball Loss on 94-03-LOOCV	96.58	91.00	86.91

2 Discussion of Performance

Average Mean Pinball Loss

Forecasting performance varies greatly by site and issue date as shown in Figure 6:

- Libby Reservoir is much more difficult to predict than the others sites. In general, sites with larger volumes have a larger pinball loss because the pinball loss is based on absolute errors.
- As expected, the closer the issue date, the lower the pinball loss. For pre-season dates it's a smoother decline compared to in-season where we have a characteristic monthly step pattern which are due to the prediction task becoming easier as parts of the seasonal volume become known.
- The 3 sites without monthly naturalized flow data (American, Merced, San Joaquin) don't show too much improvement in prediction performance during season compared to the sites with antecedent flow. It's natural that they would have less improvement than the sites with monthly flow data, but I was a bit surprised that they didn't show any relevant improvement. This would be something to look at if a solution similar to mine is adopted.
- April predictions for Libby is an outlier: The pinball loss is, on average, larger than in March. I'm not sure what the reason for that is yet.

I aggregated performance by site (averaged over the 20 years and all issue dates) and computed the Pearson correlation between site characterisitcs and average pinball loss:

- The larger a site's average seasonal volume, the harder to predict (strongly correlated 0.94).
- The larger the area of site, the harder to predict (moderately correlated 0.58).
- Higher altitude is correlated with easier prediction (weakly correlated -0.38).

Next, I aggregated performance by year (averaged by site and issue date) and computed the Pearson correlation between volume and accumulated PPT (SWANN).[3] But to give each site equal weight in the analysis, I first converted Volume and PPT to ranks (with 1 being the wettest year and 20 the driest).

- The larger the yearly volume, the harder to predict (moderately correlated 0.51)
- The larger the yearly PPT, the harder to predict (moderately correlated 0.48)

[3]Obviously, volume and yearly PPT are very strongly correlated (0.90), so results are quite similar.

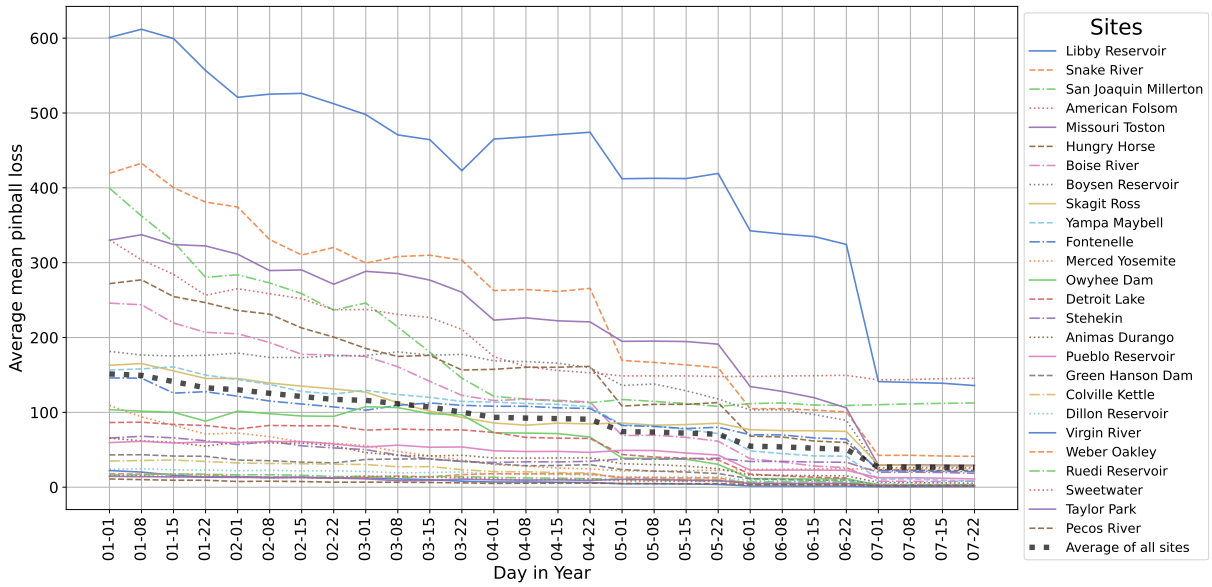

Figure 6: Each line represents the average mean pinball loss per site averaged across the out-of-fold predictions of the 2004-2023-LOOCV and in the run of issue dates.

The worst and best predictions were:

- Worst: Libby Reservoir 2012, with an average pinball loss of almost 2000
- Best: Pecos 2006, with an average pinball loss of 2.5

Coverage

I also looked into the performance and coverage aggregated by site (over the 20 years and all issue dates), see Figure 7. Most sites have a coverage of 80% or more with Detroit Lake (75%) having the lowest coverage. Sites with larger drainage areas tend to have larger errors.

3 Changes Between Stages

Hindcast model = xgboost-based conformalized quantile regression

In the hindcast stage, the prediction model consisted of 3 xgboost models, one for each quantile using the features site id, day in the year, antecedent flow, and the latest SNOTEL wteq estimate. The models were trained, tuned, and calibrated by 100 repetitions of subsampling. In each iteration the data was split into training, validation, evaluation, and calibration. A final model per quantile was trained using early stopping based on the validation splits, and the lower and upper quantiles were adjusted for coverage based on the calibration data (approach called conformalized quantile regression).

Forecast model = Hindcast model + SWANN

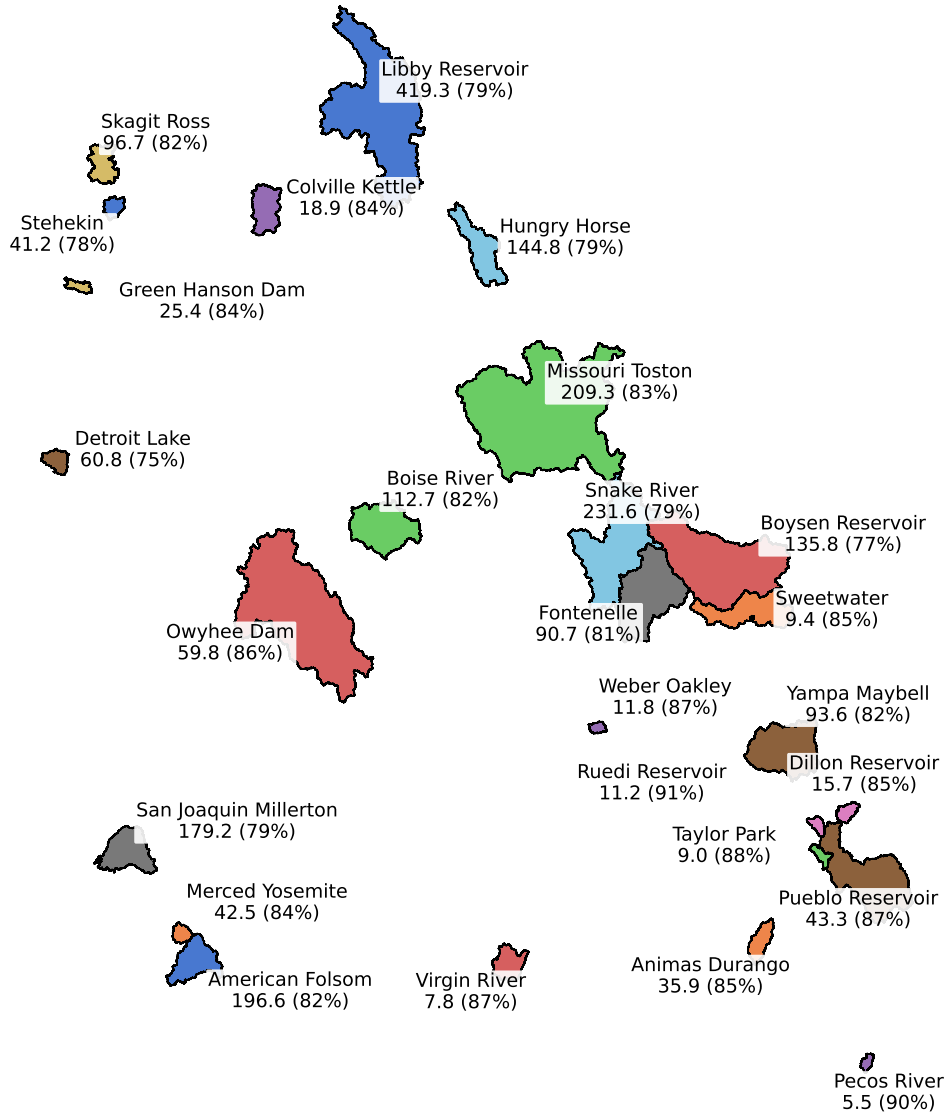

Figure 7: Average mean pinball loss and coverage by site.

In the forecast stage I made the following changes: I added features based on the SWANN data soure (swann_swe_last and swann_swe_accumulated). I also reduced the number of subsamples from 100 to 50 and maximum tree depth from 6 to 5.

Final prize stage model = Forecast model + more rigor

In the final prize stage, I made the following adjustments to my modeling approach:

- Instead of subsampling for training, tuning and calibrating the models, I use 10-fold cross-validation to train and calibrate the models. This approach is faster, uses data more efficiently, and produces better coverage results.
- Instead of fitting a final model with all data, the final prize stage solution is an ensemble of the 10 models from cross-validation (per quantile). This makes the predictions more robust.
- Lower and upper quantile adjustments are now site and issue day specific for coverage by site.
- Instead of training on all available data, I use a cutoff at 1975. This cutoff was chosen based on 94-03-LOOCV, see Figure 8. This has multiple advantages: faster modeling, better performance, and all sites are equally represented.
- Fixed the end-of-month melt bias by using conditional dates for the SNOTEL and SWANN features instead of the latest, and added feature "Unaccounted precipitation".

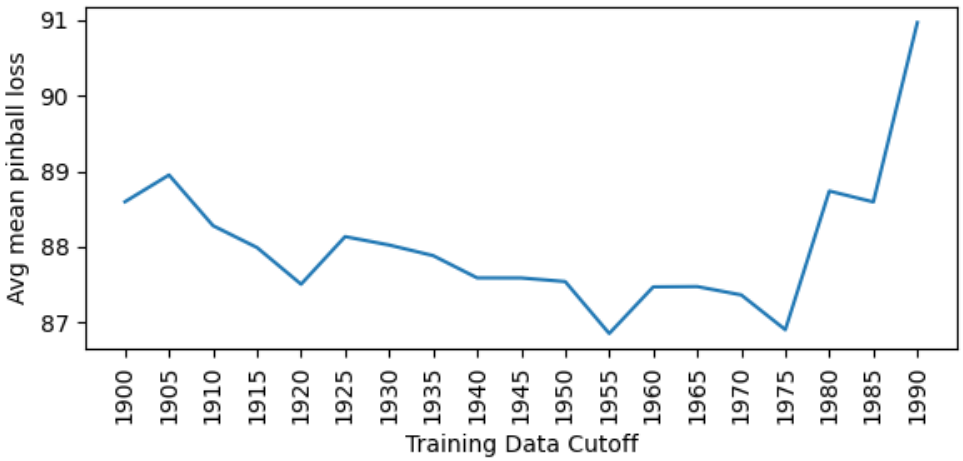

Figure 8: The average pinball loss on 94-03 LOOCV with different cutoff dates for the training data. There seems to be a tradeoff between quality and quantity: Earlier cutoff dates mean more data, but with more missing values for SNOTEL/SWANN (years before 1982). 1975 is a sweet spot.

4 Machine Specifications

All data creation, training and prediction was done on a MacBook Air M1 (2020) with 8 CPU cores and 16 GB of RAM. **Runtime needed to train the models: 6 seconds** (for all 3 x 10 xgboost models). The 20 iterations for the 2004 to 2023 LOOCV loop run in 2 minutes. Feature creation (excluding downloading) takes 4.5 minutes but this can be sped up by using merges instead of building the data row by row (a remnant from using the preprocess function of the forecast stage).

References

Breiman, Leo. 1996. "Bagging Predictors." *Machine Learning* 24 (2): 123–40. https://doi.org/10.1007/BF00058655.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. KDD '16. New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785.

Fleming, Sean W., and Angus G. Goodbody. 2019. "A Machine Learning Metasystem for Robust Probabilistic Nonlinear Regression-Based Forecasting of Seasonal Water Availability in the US West." *IEEE Access* 7: 119943–64. https://doi.org/10.1109/ACCESS.2019.2936989.

Romano, Yaniv, Evan Patterson, and Emmanuel Candes. 2019. "Conformalized Quantile Regression." *Advances in Neural Information Processing Systems* 32.

Shwartz-Ziv, Ravid, and Amitai Armon. 2022. "Tabular Data: Deep Learning Is Not All You Need." *Information Fusion* 81 (May): 84–90. https://doi.org/10.1016/j.inffus.2021.11.011.

"Statistical Techniques Used in the VIPER Water Supply Forecasting Software." 2023. Natural Resources Conservation Service. https://directives.sc.egov.usda.gov/34239.wba.