**QA Framework for Predictive Modeling**

Authors: Ellie White, Jeremy Diaz

05/19/2023

# Scope

* Objectives for producing a quality assurance plan for predictive modeling is to ensure accuracy, completeness, reliability, and efficiency and to avoid ineffective experimental designs, inappropriate analyses, and flawed reasoning.
* This document lists thoughts that are unique to predictive modeling and attempts to not repeat best practices and processes that overlap with other sections in the [QA framework](https://doimspp.sharepoint.com/:w:/r/sites/IIDDStaff/Shared%20Documents/IPDS%20and%20FSP/ITRP%20-%20Internal%20Technical%20Review%20Procedures/Draft%20QA%20Framework%20for%20Review.docx?d=w40b8bc63e4e542e4829e7237d6ec15cb&csf=1&web=1&e=k9Imwj) like version control, peer review, continuous integration and deployment etc.

# The Predictive Modeling Commandments

* Justify discarding data points.
  + E.g., we remove basins from an analysis because the watershed is considered “altered” or the gage is right below a reservoir.
  + E.g., Sometimes to avoid spurious correlations (caused by extreme/outlier data points or subgroups), we must remove data points. Reported correlations should be accompanied by a scatterplot.
  + E.g., Extremes may be present but in too low of frequency to feasibly model.
* Test **parametric** modeling assumptions – assumptions like multivariate normality, no or little multicollinearity, no auto-correlation, homoscedasticity
  + E.g., linear model residuals should show no patterns (assumption is that errors come from a random distribution) – check normality with Q-Q plots
  + E.g., linear model standardized residuals should have a constant variance (homoscedasticity)
  + Check for outliers and high-leverage points (Cook’s distance scores)
* Use a big enough sample size - otherwise analysis may be overconfident and avoid drawing conclusions based on insufficient data.
  + In the case of imbalanced classes, construct replications (with tiered re-sampling).
    - At a minimum, conduct/present evaluation at the class-level to communicate whether good or equal performance exists at important subsets of modeling.
* Avoid information leaks (cross-validation).
  + Test set should be set aside for final analysis and no alterations to the modeling framework should be allowed after the test set is applied and the model is evaluated. Best practice is to have a training – validation – test splits. Training and validation splits can be used iteratively (i.e., k-fold cross-validation being best/preferred practice), and test set is preserved for final model evaluation.
    - Class (im)balance within these partitions is important for delivering the most accurate predictions (I.e., the partitions should be similar) or answering scientific questions (I.e., how will the model generalize to differences reflected by them, such as hotter and drier days)
  + Optionally, consider what was used to calibrate/train external estimates/models that you’re using as inputs (acknowledging that this may represent a weaker data leak)
  + Good references:
    - Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera‐Arroita, G., ... & Dormann, C. F. (2017). Cross‐validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, *40*(8), 913-929. [[LINK](https://onlinelibrary.wiley.com/doi/full/10.1111/ecog.02881)]
* Set your seed once or do a replication study.
  + The more tests you run the more likely you are to encounter a false positive result (p-hacking). Report results with caution as there is always a chance the stars have aligned to give you the result you want.
  + Using a fixed/a priori set of seeds can be useful for model comparison to demonstrate variability in performance due to initial parameterization, which can sometimes be large. Likewise, set this set of seeds once at the beginning.
* Establish a process for systematically tuning model hyperparameters.
  + Consider techniques such as grid search, random search, or Bayesian optimization.
  + Step through the tires of complexity if the gains justify the effort (in most NN architectures they don’t).
* Construct appropriate evaluation metrics that align with the domain and problem.
  + E.g., In the case of streamflow droughts, should the problem be a regression type of problem predicting streamflow or a classification problem predicting the drop below a threshold.
* Model selection methods – right now, I think we generally say the latest iteration of the model is the best
* Set up a monitoring system to track the performance of ML models as they train.
  + Early stopping based on validation set performance should almost always be used to reduce resource usage and improve predictive performance while automatically tuning the number of training iterations.
* Feature selection plan – consider selection bias
  + Consider end goals of model – what would limit the timely operational delivery of model predictions? What inputs are not available through space and time?
  + In expectation-exceeding examples and as time permits, we perform deeper evaluation/explanation of features’ effect on predictions
* Consider the use of uncertainty quantification
  + A prediction without associated uncertainty is less useful. When we’re predicting 25C, preferably we elaborate that to 0-50C, 24-26C, etc for better communication, trust, and decision making/reasoning
  + Many examples have been/are being used in IIDD/PUMP/WMA (e.g., model ensembles, driver/input ensembles, monte carlo dropout, PI3NN , conditional variational autoencoder, mixture density networks, quantile regression)
  + External presenters from IBM have provided resources: Algorithms — uq360 0.1 documentation
* Model inference, issue of confounding variables
* More novel or cutting-edge modeling approaches...
  + ...should be prototyped with minimal/reduced effort simulated data at the expect size/shape of real-world data (e.g., do we have resources to run this idea/scale? How much would it (not) cost? Does the idea work on easiest/ideal conditions – e.g., known causal relationships, type of random noise, changes through time?).