Name: A. David Lander

Winner, M5 Uncertainty Forecasting Competition
First Place, Kaggle Coronavirus Forecasting Series
Overall Winner of the U.S. Department of Justice Recidivism Forecasting Challenge

Solution:

This is a classic time series forecasting situation, with only ~200 independent data points.

The core of the challenge is ensuring robust cross-validation, very high regualization, and model averaging.

This includes purged KFold cross-validation with a ~30 day gap, fold and parameter rotation across each training iteration, and some clever parameter tuning. LightGBM is the gold standard for these situations, along with parameters selected intra-fold across every regualization option.

The most notable parameter was linear_tree, which fits a linear regression inside each leaf of the tree. I've never seen this used, but it worked very well for this competition given the strong (~0.4 - 0.6 in many cases) feature correlations with the target and limited set of data points.

Models were trained independently for all locations, and features were extracted using concentric circles around each grid point (0.05, 0.1, ... 5 degree radius).

A final ensemble with a neural network added an additional 0.5% to the model's performance.

Full details are in the public source code and a broader discussion is available in a forthcoming IJF article, Forecasting with Gradient-Boosted Trees: Augmentation, Tuning and Cross-Validation Strategies.

Technical Details:

LightGBM:

- All models are fit with linear_tree, which builds a linear model inside each leaf.
- Nearly all regularization parameters are drawn from randomized search internal to each fold.
- One stretch of 0-10% of contiguous time-stamps are dropped from each training fold, and many columns are randomly dropped individually and by category.

Neural Network:

- The network is a 2-layer MLP, with three paths; each path has its own input dropout of 0.2.
- Grid_ids are embedded into 8 dimensions fed directly into the final layer.
- Models are trained with mean-squared-error loss.
- Learning rates, network size, weight decay, and gaussian noise are chosen randomly, with checkpoints stored after 5, 10, 15, and 20 epochs.

Ensemble:

- LightGBM models are trained with ~8 bags, and each neural network architecture averages ~20 bags. Each with unique folds including dropout of 0-10% contiguous timestamps from training.
- A custom model stacker is used, which minimizes mean squared error loss, while maintaining model weights of at least zero, that sum to exactly 1. L1 and L2 regularization is applied to weights.
- Models generally fit to 70% LGB and 30% NN.

Potential Improvements:

- The neural networks are undertrained, and could slightly improve.
- Running LightGBM models with exact parameter specifications, and loading them into the stacker, would likely improve performance.
- Running models leaving out an entire data source, and loading them into the stacker, would likely improve performance.